La Commode

Le Magazine des Ordinateurs Commodore Atari et Oric



n° 9
Dec. 83
45 Francs

Le sommaire complet est en page 4

Editorial

Vous trouverez des tas de choses intéressantes dans ce numéro, que votre ordinateur soit un CBM, un VIC, un 64, un ORIC ou un ATARI, ... que vous l'ayez depuis peu ou depuis longtemps déjà.

Le cours d'assembleur qui commence dans ce numéro concerne tous les systèmes nommés précédemment, qui fonctionnent tous avec le même microprocesseur 6502.

Une autre nouveauté dans ce numéro: la Programmathèque qui vous donne plusieurs programmes. Certains de ces programmes sont destinés à une seule machine, d'autres peuvent être adaptés à d'autres machines, et d'autres encore, parce qu'ils n'utilisent aucune particularité d'une machine donnée, marchent sur tous les micro-ordinateurs. Bien sûr, cette nouvelle rubrique est avant tout la vôtre: envoyez nous les programmes que vous aimeriez y trouver.

Et toujours des trucs, des astuces et des articles plus profonds, ou des tables telle que celle des adresses des routines arithmétiques pour les différents Basic.

Bonne lecture, bonnes fêtes et bonne année.

Daniel TRECOURT

Votre meilleur atout les cartes SEDERMI

a - Les cartes EXTVIC-BUS 4 / EXT64-BUS 4 qui permettent de connecter simultanément 4 cartouches ou extensions à votre VIC, par exemple VICMON, PROGRAMMER'S AID, SUPER EXPANDER et une extension mémoire, ou toute autre combinaison.

La carte EXTVIC-BUS4 existe en 3 options :

- N La carte montée et testée, avec alimentation par le VIC; c'est l'option normale.
- E Comme N, mais alimentation extérieure (non fournie). Cette option est utile pour certaines extensions d'entrées-sorties et applications particulières.
- S Permettant, grâce à 4 interrupteurs d'activer la cartouche voulue et ainsi d'avoir 4 cartouches de jeu ou Super Expander montées en permanence.

b - Les cartes VIA 2 (pour VIC, 64, ORIC) qui procurent 40 lignes d'entrées-sorties supplémentaires (idéal pour commander des appareils domestiques, des trains électriques etc.)

Chaque carte contient deux VIA 6522. Il y a 4 adresses de base au choix. On peut connecter simultanément 4 cartes VIA 2.

Existe en version 1 VIA pour ORIC.

c - La carte EXTVIC-EPROM 4 qui contient 4 supports d'EPROM 2532, ce qui vous permet d'avoir vos programmes permanents figés en EPROM et disponibles dès la mise sous tension.
Chaque EPROM peut être activée ou inhibée par interrupteur et être implantée au choix en 4 adresses.

d - L'interface ORIC-JOY 1 qui permet de connecter un manche à balai Commodore, Atari ou compatible à votre ORIC pour 65F.

En préparation, cartes relais, programmateur d'EPROM.

BON DE COMMANDE :

à adresser avec le règlement à SEDERMI, 28, rue Vicq d'Azir, 75010 PARIS
(délai à prévoir : un mois environ)

Nom : ....

Adresse : ....

....

```
......ORIC-JOY 1     ( 65F TTC) :        :
......EXTVIC-BUS4-S  (480F TTC) :        :    règlement par
......EXTVIC-BUS4-N  (390F TTC) :        :
......EXTVIC-BUS4-E  (390F TTC) :        :       CCP : ()
......EXT64-BUS4-S   (480F TTC) :        :
......EXTVIC-VIA2    (690F TTC) :        :       CB  : ()
......EXT64-VIA2     (690F TTC) :        :
......EXTORIC-VIA2   (690F TTC) :        :     mandat : ()
......EXTORIC-VIA1   (510F TTC) :        :
......EXTVIC-EPROM4  (310F TTC) :        :
          TOTAL ...........              F TTC
```

Sommaire n° 9

Tous à Villemomble !

Le Club Informatique 20 - 64 organise les 21 et 22 janvier 1984 un week-end informatique dans le préau de l'école Foch, 73 rue Bernard Gante à VILLEMOMBLE 93250

Seront présents:

Plus de 40 VIC 20 et Commodore 64 avec tous leurs périphériques,

Les derniers périphériques sortis,

Plus de 900 logiciels de la bande dessinée à la gestion en passant par les jeux,

Imprimante - traceur de courbes,

Table traçante et table à digitaliser,

Des produits étrangers,

Démonstration des possibilités du Langage Machine et du FORTH,

Démonstration de Base de données,

Liaison de 6 Commodore 64 à une même disquette 1541 et même imprimante,

Et des nouveautés encore non commercialisées ...

Courrier des lecteurs

64 graphique

Je possède un CBM 64.

1 . Peut-on protéger un programme d'une façon ou d'une autre ?
2 . Quelques adresses pour la génération de caractères me seraient utiles.
3 . Comment procéder pour faire un dessin (sans sprites) en haute définition ?
4 . Je souhaiterais quelques indications sur la programmation du 64 en assembleur.
5 . Une cassette comme "UTIL-VIC" pour 64 serait vraiment utile !
6 . Est-ce que M. Daniel-Jean DAVID va écrire un livre sur le 64 ?

Olivier JAUNIN
1007 Lausanne

1 . Tout dépend contre quoi protéger. Contre la copie : il existe differentes méthodes qui seront abordées par La Commode.

2-3. Questions traitées dans la "Découverte du 64" éd. du PSI.

4 . Dans ce numéro, nous commençons notre série sur l'apprentissage de l'assembleur.

5 . Nous croyons savoir que H. Le Marchand y pense.

6 . "La découverte du 64" est parue au SICOB. "La Pratique" volume 1 est prévue pour novembre ou décembre, donc devrait-être parue au moment où vous recevrez votre n° 9.

Saut de ligne et RS232 du VIC

Je viens de me porter récemment acquéreur d'un VIC 20. Etant électronicien et ayant à ma disposition des imprimantes diverses à entrée RS 232, j'ai réalisé une interface pour passer de 0..+5V à -12..+12V pour attaquer mon imprimante. Tout marche (110 ou 300 Bauds) quand je veux lister mes programmes, sauf que je n'ai pas d'avance papier ! (Line feed)

Y a-t-il un truc pour incorporer automatiquement cette fameuse instruction CHR$(10) à chaque retour chariot (CHR$ (13)) ? Je n'ai une avance de papier qu'à la fin de mon listing, pour le READY ! C'est peu.

D'autre part, où pourrais-je trouver le schéma complet du VIC ?

Je viens de m'abonner à LA COMMODE et de commander les numéros qui manquent. Peut-être la réponse à ma question se trouve dans un des numéros ? En tout cas, bravo pour cette revue passionnante et indispensable.

P.S. Les cartes RAM remplies sur option sont-elles sorties ?

J.C. VARLET
31450 MONTLAUR

Votre problème réside plutôt du côté de l'imprimante : il faudrait qu'elle fasse une avance de ligne en recevant le code 13. Une solution hardware de transformation du code 13 en code 10 + 13 est très difficile en interface série.

Sinon, il faudrait réécrire la routine de LIST pour qu'elle envoie un CHR$ (10) en fin de ligne. C'est assez délicat.

La décision de faire des cartes RAM n'est pas encore prise à la SEDERMI.

Mode multicouleurs et extensions du VIC

Pouvez-vous m'aider à éclaircir quelques points à propos du VIC ?

- Le mode "multicouleurs" : quand on poke en RAM couleur un nombre supérieur à 7, on obtient un emplacement 8 x 8 non uniformément coloré. Mais, qu'elle est l'action exacte de la valeur pokée sur le coloriage de l'emplacement ?

- Peut-on installer dans un boitier d'extension mémoire, en le modifiant, des puces afin d'avoir les 27,5 Ko de BASIC et les 3 Ko de la zone mémoire de l'extension 3 Ko (adressables en machine uniquement dans ce cas) disponibles sur le VIC. Sinon, pouvez-vous nous donner le schéma nécessaire pour avoir toute cette mémoire ?

- D'autre part "qu'y a-t-il" aux adresses mémoires suivantes :

de $9200 à $93FF avec les grosses extensions
de $9200 à $95FF avec une RAM (= 6,5 Ko utilisateur, c'est-à-dire la mémoire comprise entre les adresses d'E/S et la RAM couleur.

de $9600 ou $9800 (juste après la RAM couleur)
jusqu'à $BFFF (juste avant la la ROM du BASIC)

Ceci fait, lorsque je rentre

```
$1000 = RAM d'écran (début)
$2000
LDY    $ FF
BOUCLE : TYA
STA $ 1000 Y
DEY
BRK
```

avec l'extension 8 Ko en place ,le VIC se "plante" au point que RUN/STOP- RESTORE est inefficace et je dois obligatoirement éteindre le VIC.

Une toute dernière chose: vous dites dans le n° 6, en réponse à la lettre d'Alain DRISSENS, à propos des "Touches programmables du VIC", d'aller voir dans "VIC à BRAC" du même numéro, mais où ?
Il n'y a rien de cela dans cette rubrique !

Cette remarque étant faite en toute sympathie, Bravo à tous et à quand un numéro bimestriel?

Daniel JALLAIS
78120 ORPHIN

J'ai reçu récemment la carte EXTVIC-BUS 4 N que je vous avais commandée.

J'ai voulu l'utiliser en branchant dans le VIC 20 les cartouches SUPER EXpander + 16 k. J'ai obtenu 19 Ko de mémoire. Or, j'ai un programme avec fichier et graphismes qui comporte un DIM CV (50,58) qui refuse de fonctionner avec seulement 19 Ko de mémoire.

Je me suis donc procuré une mémoire 8 Ko que j'ai ajoutée, mais, à ma grande déception, je n'ai pas obtenu de mémoire supplémentaire.

Votre notice de la carte EXT-VIC BUS 4 indique comme incompabilités : RAM 8 K et RAM 16 K tant qu'on n'a pas rendu leurs espaces d'adressage non recouvrants.

Michel MANET
75007 PARIS

1) Le mode multicouleurs sera explique dans un prochain n° de LA COMMODE. Il a en fait assez peu d'intérêt.

2) Pour avoir 27 K, il faut modifier des interrupteurs dans la cartouche 8 K, comme c'est explique dans le n° 8.

3) Il n'y a rien aux adresses $9200 à 95FF, ni de 9800 à 9FFF. En revanche, de A000 à BFFF il peut y avoir des ROM d'extension comme SUPER EXPANDER, MASTER, etc.,

4) Votre plantage ne s'explique pas.

5) L'explication des touches de fonction a été squeezée. Elle a été donnée depuis dans "Clefs pour le VIC".

Compatibilite VIC et 64

1) Y a-t-il un ouvrage sorti concernant le COMMODORE 64 ou quand est-il prévu ? (Je n'ai que le manuel du 64 en anglais)

2) Schéma branchement sortie audio-vidéo DIN - entrée prise 21 broches PERITEL pour VIC 20 et 64?

3) A votre cnnaissance, y a-t-il une possibilité de rendre compatibles des programmes VIC 20 mis en K7, afin de les utiliser sur le 64 ?

4) Est-ce que "Clefs pour le PET/CBM" peut servir pour le 64 ? Est-il prévu un mémento particulier pour le 64 ?

Maurice ALLEMANE
47000 AGEN

1) Vous devriez avoir la version française du manuel PROCEP du 64.

La "découverte du 64" de DJD est parue au SICOB. Les autres paraîtront après le SICOB.

2) La liaison PERITEL du 64 est différente de celle du VIC. Le schéma a été donné dans le numéro 8 de LA COMMODE.

3) Le 64 peut lire des cassettes de VIC.Il y a les problèmes habituels d'affichage et d'adresses.

4) Il y aura un "Clefs pour le 64".

STATUS ... quo

Dans le manuel d'utilisation, il y a mentionné la réservation par le système de la variable STATUS : son contenu indique la nature du problème de chargment à partir d'une cassette, lorsqu'il y a lieu.

Pouvez-vous me dire la signification de chaque valeur de cette variable ?

Olivier BUTLER
94420 LE PLESSIS-TREVISE

Pour ST, vous pouvez consulter "Clés pour le VIC" aux éditions du PSI.

TO DRAW, DREW, DRAWN ...

J'ai cherché à tracer des courbes et dessiner des graphiques en haute résolution sur VIC, mais je me trouve confronté au fait que tous les points ne sont pas jointifs et qu'entre quelques uns de ceux-ci, j'ai des blancs, pourriez-vous m'indiquer la solution pour que tous les points soients jointifs, me donner les références d'un livre ou d'une revue qui traite de ce problème. Merci.

Hervé DABBADIE
93160 NOISY-LE-GRAND

La solution la plus facile de votre problème consiste à avoir la cartouche SUPER EXPANDER et utiliser la commande DRAW. Sinon, il faut simuler cette commande par un programme délicat donné dans ce numéro.

Utilitaires et extensions memoire du VIC

Pourriez-vous m'apporter une aide dans les problèmes qui me sont posés par mon ordinateur :

1) Je suis possesseur d'un VIC 20 avec la cartouche "SUPER EXPANDER" et j'aimerais acheter la cartouche 32K RAM commercialisée par RUN informatique (7-11, rue de l'Yvette, 75016 PARIS). Cette cartouche est commutable en 16K ou 32K. Dans l'option 32K cette cartouche ne va-t-elle pas se superposer avec le "PROGRAMMER'S AID" et le "VIC-MON" vers le haut de la mémoire. D'autre

part, avec la mémoire de base, plus l'extension 3K il reste 8K (sur 32K) qui ne sont pas utilisables. La publicité dit que ces 8K sont utilisables en langage machine: est-ce possible ?

2) L'alimentation du VIC est-elle suffisante pour alimenter la carte SEDERMI "EXT VIC BUS4" avec "SUPER EXPANDER", "PROGRAMMER'S AID", "VIC MON" et une cartouche 16K ou la cartouche 32K citée plus haut.

Pierre-Guy ALLINEI
04160 CHATEAU-ARNOUX

1) La cartouche 32 K RAM a 8K en A000-BFFF qui sont incompatibles avec SUPER EXPANDER et les jeux mais accessibles en L.M. Avec VIC-MON (6000...) et PROGRAMMER'S AID vous devez utiliser la cartouche en mode 16K.

2) L'alimentation du VIC est parfaitement suffisante pour SUPER EXPANDER, VICMON, PROGRAMMER'S AID et + une 16K ou la 32K (à utiliser en mode 16K).

FORTH

J'ai eu un compilateur FORTH 2.0 de CAUBOIS SOFTWARE (C) 1981. N'ayant pas la notice, j'ai tenté d'analyser le compilateur avec le livre de Léo BRODIE "STARTING FORTH". J'ai découvert 50% des instructions. Aussi, je serai particulièrement intéressé par tout renseignement sur le FORTH pour CBM série 3000 et surtout sur cette version FORTH 2.0.

Connaissez-vous d'autres versions de FORTH adaptables sur CBM 3000 commercialisées en France ?

André HESBOIS
94800 VILLEJUIF

Nous ne connaissons pas la version de FORTH 2.0. Nous connaissons FORTH 4000/8000 de DATATRONIC qui marche très bien et qui est (?) commercialisé par PROCEP, mais dont la version 3000 n'est pas venue en France à notre connaissance.

"Gestion de fichier" sur VIC

M. BRUGEASSOU a écrit un PRG dans le N° 5 de LA COMMODE, qui me passionne énormément et ouvre quantité de possibilités (PRG: gestion de fichier).

Après pas mal de recherches, je ne suis pas parvenu à l'adapter pour un VIC 20, du moins pour ce qui concerne l'incrémentation du compteur DATA (ligne 2025) : à quelle adresse? (1020 ?)

- que devient CHR$ (34) sur un VIC?

- que deviennent les POKE ? (lignes 2060 - 3020).

M. KEULERS

La solution de votre problème réside p. 35 dans le N° 6 de LA COMMODE. Les CHR$ (34) sont inchangés.

En 50 158 --> 198; 1010, 1040 152 --> 653 2060, 3020 158 --> 198 ; 623, 624 --> 631,632

Passage du 3000 au 64

Les possesseurs de 3000 doivent actuellement s'interroger sur le 64. En ce qui me concerne les questions sont nombreuses :

- Y aura-t-il compatibilité avec les périphériques possédés; pour mon compte, il s'agit des 4022 et 2031. (L'option bus IEEE devrait sans doute résoudre le problème).

- Qu'en est-il de ce programme qui "émulerait" le PET ? Nos programmes en langage machine pourront-ils être récupérés ? (J'ai acheté le traitement de texte de PROCEP, Extramon, de nombreux jeux).
Et tous les programmes rédigés avec l'EDEX 2.0 ?

- Reste la question du clavier AZERTY et surtout des CARACTERES

ACCENTUES, ont-ils des chances de fleurir sur le 64 ?

Jacques BOUTIQUE
87000 LIMOGES

La compatibilité du 64 avec les périphériques du 3000 est assurée par l'interface IEEE. Nous avons essayé celle de DAM'S (G.B). Elle est diffusée en France par PROCEP (950 F).

Nous n'avons pas encore essayé le programme d'émulation PET. Il n'est pas évident que tous les programmes soient utilisables. Maintenant, le mode émulation empêche de bénéficier des avantages de la nouvelle machine.

Le traitement de textes 64 est très bon. Il devrait être francisé.

64 : documentation et publicite

Je me permets de vous écrire pour vous informer que j'ai suivi votre conseil et j'ai acheté un COMMODORE 64.

Cet ordinateur me semble parfait pour un usage domestique et permet un nombre important de possibilités.

Je ne suis pas assez savant pour critiquer un tel appareil; toutefois, je dois vous donner mes premières impressions ainsi que mes désirs les plus urgents :

1) Il n'a été mentionné dans aucun article ou publicité que la notice n'était pas encore traduite, ce qui est très gênant pour qui ignore l'anglais (notamment pour les sprites et les POKE du son).

2) Il est mentionné partout que le son est d'une qualité extrême, soit ..., mais dans la pratique ce n'est pas le cas : volume TV sans bruit de fond, presque pas de son "normal", mais bruit de fond important (jusqu'à "entendre" le fonctionnement de 64 !! Je pense qu'une Péritel devrait résoudre le problème.

3) Mais dans ce cas, pourquoi la prise Péritel du VIC que j'utilisais ne peut pas fonctionner ? Il y aurait pu avoir compatibilité ...

4) Parlons compatibilité justement. Il est écrit que les programmes PET sont compatibles avec le 64; or, il a été dit la même chose pour le VIC donc déduction : ceux du VIC doivent tourner sur le 64. Il n'en est rien, ce qui est logique puisque ni la couleur, ni le son et ni les adresses ne sont identiques.

5) Justement, pour les adresses, sur quoi se baser pour "traduire" un programme VIC pour tourner sur 64 ? Par exemple : CO = 30720 pour le VIC qu'en est-il pour le 64 ? ou encore POKE pour créer des caractères graphiques.

Fernand BOUGON
76140 PETIT-QUEVILLY

1) La notice du 64 existe maintenant en français et vous devriez l'obtenir auprès de votre distributeur. Nous aimerions d'ailleurs que PROCEP ait pour politique que les notices traduites tardivement soient fournies de droit aux acquéreurs de la première heure (c'est après tout eux qui font vivre PROCEP) et que cette politique soit notifiée clairement aux distributeurs.

2) Les problèmes de bruit de fond sont réels et la prise PERITEL améliore les choses théoriquement, mais c'est à vérifier pour chaque couple 64 - téléviseur.

3) La connexion 64-PERITEL a été décrite dans le n° 8. Voir aussi 64 à BRAC dans ce numéro.

4 - 5) Les réponses à ces questions seront petit à petit dans LA COMMODE. Voyez aussi "La découverte du 64" aux éditions du PSI.

"feroce" 64

Possédant un VIC 20 de 16 Ko, je dois vous dire qe je suis très déçu par COMMODORE. Car ayant acheté le VIC en raison de ses qualités et de sa jeunesse (jeunesse qui garantissait une technologie de poin-

te), j'apprends par le truchement de votre journal, l'arrivée du 64. Or ce VIC 64 possède d'énormes possibilités et son rapport qualité-prix m'apparaît bien supérieur à celui du VIC 20, puisque avec ses 38 Ko de mémoire vive il coûte 2900 F, alors qu'une extension 16 K VIC 20 vaut 700 F. Aussi me suis-je livré à un calcul très simple :

VIC 3,5 Ko : 1500
2 extensions 16 Ko : 2 x 700
Total : 2900

Tout cela pour montrer qu'un VIC 20 avec 35,5 Ko coûte 2900 F, c'est-à-dire le même prix qu'un 64 avec, ô combien, plus de possibilités.

Tout ceci pour dire que les possesseurs de VIC 20 ont été dupés. "Le VIC 20 aura constitué une étape intéressante et sympathique" affirmiez-vous dans le n0 6; rendez-vous compte qu'implicitement vous affirmez : le VIC 20 est bon à mettre au rebus !

Mais, la plupart des gens qui possèdent ce micro-ordinateur sont des "jeunes", c'est-à-dire des gens pour lesquels cet achat a constitué un investissement important. Et le million de "Vicistes 20" ne veut point être lésé. Aussi, je vous pose la question suivante :

Peut-on transformer un VIC 20 en 64 et à quel prix ?

Une dernière chose: La production de VIC 20 ne va-t-elle pas être abandonnée face au "féroce" 64?

Christian LAPORTE
62200 BOULOGNE-SUR-MER

1) Il n'est pas possible de transformer un VIC en 64 autrement qu'en vendant le VIC d'occasion (il devrait y avoir un marché actif) et en achetant un 64.

2) Nous ne pensons pas que les acheteurs de VIC ont été dupés :

- le 64 est apparu bien après et vous avez profité du VIC dans l'intervalle : si l'on attend la prochaine machine qui sera bien mieux, on n'achète jamais rien et on n'a rien.

- le VIC conservera toujours ses applications et son marché; à notre avis, il ne sera pas abandonné de sitôt.



Tampon clavier et touches programmables

Puisque M. D.J. DAVID nous invite dans le n°4 à faire part des applications utilisant le tampon-clavier, j'en propose une :

Supposons que nous ayons écrit un petit programme en langage-machine à partir de l'adresse 826 et que nous souhaitions le sauvegarder en Basic. Il est possible de faire créer les lignes de DATA par l'ordinateur lui-même grâce au petit programme suivant :

```
10 PRINT "Clr 1000 DATA";
20 FOR I = 0 TO 9
30 PRINT PEEK (826 + I);
40 PRINT "," ;
50 NEXT
60 PRINT CHR$ (20) : REM SUPPRES
SION DERNIERE VIRGULE
70 POKE 158,2:POKE 623,ASC("HOME")
: POKE 624,13.
```

Il n'y a plus ensuite qu'à faire "LIST" pour voir que la ligne 1000 est venue s'ajouter au programme. On peut ensuite créer la ligne 1010 de la même façon: en 30, il faudra remplacer 826 par 836.

Je vous remercie, par ailleurs, d'avoir signalé l'ouvrage de B. MICHEL : "Routines internes du PET-CBM". Commandé en Belgique, il m'est parvenu à la Réunion très rapidement et j'y ai trouvé des choses très intéressantes. En particulier, l'astuce qui consiste à détourner la routine d'interruption vers l'éternelle adresse 826 pour y placer un programme qui sera exécuté tous les soixantièmes de seconde. L'auteur donne deux exemples : la répétition de touche et la façon de rendre la touche "RUN STOP" utilisable même en langage-machine. Après avoir bien étudié ces 2 programmes, j'ai eu l'idée de faire une touche programmable. J'ai choisi pour cela la touche "&" qui ne sert pratiquement jamais et j'ai fait le programme suivant :

```
100 REM *** PROGRAMMATION TOUCHE
    '&' ***
110 :
120 REM : NE JAMAIS FAIRE SYS 826
130 FOR I = 826 TO 859
140 READ A : POKE I,A
150 NEXT I
160 DATA 165, 151, 234, 201, 77,
    240, 4, 169, 254, 133, 0, 230,
    0, 240, 3, 76,
170 DATA 85, 228, 169, 112, 160, 3,
    32, 29, 187, 76, 85, 228, 22,
    253, 76, 198,
180 DATA 183, 96
190 :
200 : REM *** MESSAGE ***
210 :
220 CH$ = "TOUCHE PROGRAMMEE"
230 CH$ = CHR$ (20) + CH$
240 :
250 FOR I = 1 TO LEN (CH$)
260 POKE 879+I,ASC(MID$(CH$,I, 1))
270 NEXT I
280 POKE 880 + LEN (CH$),0
290 :
300 REM * DETOURNEMENT
    INTERRUPTIONS VERS 826 **
320:
330 DATA 120, 169, 3, 133, 145,
    169, 58, 133, 144, 88, 96
340 FOR I = 1000 TO 1010 : READ B:
    POKE I,B : NEXT I
350 SYS 1000
```

(Si on est très paresseux, on peut remplacer les cinq dernières lignes par POKE 144, 58 : POKE 145,3. On risque un blocage, mais ça marche une fois sur deux.)

Et voilà, un petit RUN et chaque fois que l'on appuiera sur la touche "&", on verra s'inscrire "TOUCHE PROGRAMMEE". Rien n'empêche, bien sûr de remplacer ce message par un autre : il suffit de modifier la ligne 220. Ceux qui

prendront la peine de désassembler de 826 à 859 verront aussi que rien n'empêche de programmer plusieurs touches, en attribuant par exemple un mot-basic à chacune d'elles. Ainsi la touche "&" donnerait "INPUT", la touche "*" donnerait "FOR I = 1 TO", etc., Ainsi les possesseurs d'un petit PET-CBM n'auront plus rien à envier aux possesseurs d'un ZX 81 !!

Enfin, je voudrais vous faire part de la déception que me cause le langage PASCAL ou plutôt la déception que me causent les difficultés que j'ai à l'apprendre. L'absence du mot "STRING" fait que 9 sur 10 des programmes que je rencontre ici ou là ne donnent rien. J'en suis à mon cinquième livre acheté pour arriver à écrire un petit programme moi-même et si j'ai enfin réussi à faire un "GET", je n'ai pas encore trouvé le moyen de sortir d'un "INPUT". Ne pourriez-vous pas de temps à autre donner quelques tuyaux à ce sujet ?

Quel beau langage, par contre, que le COMAL ! On écrit un programme dans un Basic à peine adapté et quand on fait "LIST", c'est pratiquement du PASCAL.
J'écris par exemple :

```
10 REM ESSAI
20 FOR X = 1 TO 5
 30 LET Y = X^ 2
40 PRINT Y
50 NEXT X
```

et j'obtiens :

```
0010 // ESSAI
0020 FOR X : = 1 TO 5 DO
0030       Y : = X^ 2
0040      PRINT Y
0050 NEXT X
```

avec les "DO", les ":=" et l'indentation en prime ! J'apprécie que le langage me permette de garder quelques habitudes BASIC et qu'il se charge de les adapter. Grâce à lui, on s'habitue peu à peu au PASCAL. Un seul regret : qu'il n'accepte pas les caractères shiftés : que de fois il faut réécrire "LIST" parce que "L SHIFT I" est inconnu. COMAL veut-il bien dire "COMMON ALGORITHMIC LANGUAGE" ou comme je l'ai lu quelque part "COMMODORE ALGORITHMIC LANGUAGE" ? Même si le COMAL n'est pas une invention purement commodorienne, c'est une très belle chose.

R. BASSABER
97490 SAINTE CLOTILDE

La technique de la touche programmable est adaptable aux touches F du VIC ou du 64.

COMAL veut dire COMMON ALGORITHMIC LANGUAGE.

Reussissez aux Echecs

Au sujet de la lettre de D. DORDAIN (La Commode n°6, page 13) demandant des renseignements sur des programmes d'échecs, je vous signale l'existence des programmes suivants :

1) PET CHESS pour PET/CBM 2000, 3000, 8000,(avec 12 niveaux et un excellent graphisme), de

ACT MICROSOFT Ltd
Act House
111, Hagley Rd
Edgebaston
BIRMINGHAM 16, Angleterre
Prix : 25 Livres
Tél : (021) 454-8585

2) 420 Schach für CBM+PET 2000/3000 de :
W. HOFACKER GmbH
Tegernseerstr. 18
8150 HOLZKIRCHEN RFA
Prix : DM 79
Tél : (08024) 7331

Robert W. HAAS
83330 St NAZAIRE-LES-EYMES

Un plantage avec EDEX 2.0

Pour (l'ancienne) rubrique PETER JOLLYBUG :

Concerne : EDEX 2.0
Constatation : plantage du CBM lors d'un RENU :
D'autant plus méchant que le RENU dure 1 minute pour 500 lignes.

Essai :

```
10 PRINT "bonjour"
20 PRINT "c'est moi"
30 GOSUB 100:
40 PRINT "plante"
50 END
100 PRINT "le PET" : RETURN
```

RESULTAT :

Eliminez d'office les ":" derrière un GOSUB. L'EDEX les prend comme une instruction "vide" et fait disparaître 40.

QUESTION aux lecteurs:

Comment trouver les REM avec l'instruction FIND ?

Yves THIBAUT
Club CBM 80
AMIENS

Compatibilite EDEX-SUPERBASIC

Après nous avoir signalé, dans une première lettre, certaines incompatibilités entre EDEX et SUPERBASIC (Voir l'essai dans le n° 6, page 21), M. CHAUFFOUR nous donne la solution dans sa seconde lettre:

PREPARATION DU PROGRAMME EXISTANT ECRIT SOUS EDEX:
PERMET LECTURE RAPIDE SOUS SUPERBASIC (supprimé à l'exécution) :

1) On adapte le programme principal (voir fin de page) et on détermine les pointeurs début et fin de la liste BASIC de ce programme par

```
PRINT   PEEK(40),PEEK(41),PEEK(42),
PEEK(43)
```

2) On traduit les 2 valeurs en hexadécimal (pour 6°)

3) On éteint le CBM. Puis on rallume.

4) On charge en début le programme suivant :

```
0 SYS40960:POKE42,0:POKE43,27:
POKE623,33:POKE624,82:POKE625,73:
POKE626,78:POKE627,13:POKE628,82:
POKE629,213:POKE630,13:POKE158,8:
END
```

Utilisation du tampon clavier pour lancer la lecture rapide du programme principal qui suit. On retire la cassette sans rebobiner.

5) On fait NEW et on charge le programme principal à partir de la cassette d'origine.

6) On fait SYS 40960 (pour CBM 3000 en $A000), puis on met la cassette utilisée au 4°)
! ROUT"NOM", XXXX, YYYY (début et fin du programme trouvés ci-dessus, en hexa sans $).

L'enregistrement du programme se fait en écriture rapide.

7) Voir "UTILISATION"

```
0 !RESET  ==]   on supprime SUPER
   BASIC
100 SYS46000:PRINT""  ==]   on
   active EDEX
110 REM' JOINT PAR PLATINE
   POUTRE/POTEAU'
120 PRINT"**JOINT POUTRE/POTEAU
   (PLAT/HR)**":PRINT
130 PRINT"NFP 22-460" (programme
   principal)
```

UTILISATION DE LA CASSETTE PROGRAMMEE

Avec "RUN/STOP" on lance normalement la lecture de la cassette, avec lecture du premier programme sous CBM BASIC

```
0 SYS40960:POKE42,0:POKE43,27:
POKE623,33:POKE624,82:POKE625,73:
POKE626,78

1 POKE 627,13:POKE628,82:POKE
629,213:POKE630,13:POKE158,8:END
```

Le chargement du programme principal par SUPERBASIC avec lecture rapide et AUTORUN

```
0 !RESET (on supprime SUPERBASIC)
100 SYS46000:PRINT"" (On lance
   EDEX)
110 REM' JOINT PAR PLATINE POUTRE/
   POTEAU'
120 PRINT"**JOINT POUTRE/POTEAU
   (PLAT/HR)**":PRINT
130 PRINT "NFP 22-460"
140 PRINT"TYPE DE JOINT ?"
```

"
"
PROGRAMME
PRINCIPAL
SOUS EDEX
"
"

REMARQUE : utilisation directe avec une seule touche sur clavier + touche du magnéto.

Jean-Pierre CHAUFFOUR
46100 FIGEAC

Haute Resolution

Je voudrais faire une petite remarque au sujet de l'affichage haute résolution. Je suppose, si j'ai bien saisi le fonctionnement de la carte, que l'emploi du PORT A, pour une autre activité, risque de perturber l'affichage de l'écran par des changements de niveau sur PA3.

Il est judicieux de prévoir un interrupteur comme suit:

+5V
PA3
BR/$\overline{HR}$

De cette manière, le PORT A reste utilisable, en basse résolution non-programmée !

Michel DURAND
B 1000 BRUXELLES

Attention aux demos familiales

Tout d'abord, merci pour votre revue. Je me présente : GAUJARD Philippe, célibataire, un enfant : "VIC 20" -Age : 2 mois.

Rentrant le programme suivant qui tourne très bien à la main, voici un extrait de ce programme :

```
180 INPUT A
200 A1 = A1 + A
240 INPUT B
250 IF B = 0 GOTO 280
260 B1 = B1 + B
270 GOTO 240
280 N1 = A1 - B1
310 INPUT D
320 IF D = 0 GOTO 350
330 D1 = D1 + D
340 GOTO 310
350 PRINT
370 INPUT E
380 IF E = 0 GOTO 410
390 E1 = E1 + E
400 GOTO 370
410 PRINT
430 INPUT F
440 IF F = 0 GOTO 470
450 F1 = F1 + F
470 G1 = D1 + E1 + F1
480 IF G1    N1 THEN PRINT
    "ERREUR" .
```

Ayant fait plusieurs essais concluants, tout fier de mon savoir récent et de ma boîte "mystérieuse", pour certains dont j'étais, je convoque ...la famille. On allait voir ce que l'on allait voir! : Allumage, Load, et Run .

Me voici rentrant mes données avec à la clé les explications d'usage :
A = 198.87 RETURN
"ET A1 QU'EN FAIS-TU" m'ont-ils dit;
"C'EST LE PROGRAMME QUI S'EN CHARGE" je réponds;
B = 16,70 RETURN ? 0 RETURN
"ET B1" voir plus haut, commentaires A1 et pour toutes données en ...1. Dieu qu'il fait soif; la bouteille d'eau en a pris un coup.
D = 114.82 RETURN ? 0 RETURN
E = 67.35 .....
F = 0 ......
"ET MAINTENANT"....."Il compare N1 à la somme des 3 données D,E,F" Je ... Edition sur l'écran de A1, B1, N1, D1, E1, F1, 5 séries, 10 séries, ça marche toujours devant le public ébahi (Pas tous, certains ayant vaqué à d'autres occupations plus terre-à-terre) (ignare, va !) et alors ... "CRAC" faites vos jeux, rien ne va plus
A = 173.71 B = 25.60 E= 148.11
"ERREUR" sur l'écran et en gros (pour attirer l'attention ...) Appui sur la touche adéquate (Il y a un Get ... pour répartir, pour effacer certaines données sans toucher à d'autres) et je recommence la même série. Ah ! mais, alors ! RE -"ERREUR" sortie de la poquette familiale (moins de 100 FF une

honte) Verdict sans appel N1 = G1. Vérification du listing écrit. List au clavier. Le rouge aux joues : "L'essai n'est pas très concluant, je vais chercher l'erreur". Tout le monde repart en riant sous cape.

Après pas mal de tâtonnements de RUN STOP CONT PRINT comme s'il en pleuvait (Tiens, aujourd'hui il pleut) :

Voici la solution adoptée :

SPG arrondi 20 décimales après chaque calcul et allons-y des GOSUB et autre RETURN
DEF FNA(T) = INT(T*100 + .5) / 100

Cette fois-ci, prudence, le clairon reste au placard

A = 173.71 B = 25.60 E = 148.11
EUREKA, ça marche. C'était crier victoire trop tôt.
A = 587.47 B = 44.90 E = 542.57
"ERREUR" Ah non ! Nom de ...M.... L'informatique, ras le bol .B... (Censuré, il y a des jeunes qui lisent La Commode), mais le virus est tenace, et après des heures et des nuits (on travaille le lendemain) INT(T*100 + . 5) / 100 devient : INT(T*100 + . 4) / 100
Pourquoi ? J'ai peut-être un début d'explication en vous disant que sur le VIC en ma possession (.5) - (.4) = .1 mais (.7)-(.6) = .0999999 Maths modernes et informatiques tout de même sont étranges.

2) Eh oui, ça continue (pire que Dallas) : Ouvrons un fichier pour stocker mes résultats (chèrement acquis, heureusement qu'il y a les dimanches et pas mal de jours fériés). Prenons le mode d'emploi.

Est-ce qu'il ressemble à un mode d'emploi ? Enfin, passons.

Ecriture 1:

```
OPEN 1,1,1,"FICH"
PRINT 1,liste variables
CLOSE 1
```

Lecture 1:

```
OPEN 1,1,0,"FICH"
INPUT 1,liste variables
```

Dans ma petite tête cartésienne liste des variables = A, B, N1, D1, E,. Et allons-y gaiement. A l'enregistrement, tout a l'air de se passer normalement.

Rebobinage cassette. Lecture. Alors là ! Désespoir:

SYNTAX ERROR IN ...
STRING TOO LONG

J'en passe et des meilleures mais pas tout à la fois, quand même.Tiens "La Découverte du VIC" vient d'arriver. Mais, hélas ! Fichiers : voir au prochain numéro. Quand svp?

"Le guide du programmeur VIC" en anglais svp (Il s'agit, le VIC, d'un matériel américain fabriqué en Allemagne avec sûrement des composants japonais, utilisé par des français. Allez dire que l'informatique ne fait rien pour le rapprochement des peuples...). Commande chez PROCEP (vous connaissez): doit arriver dans 2 mois. Bien, alors expérimentons :

Ecriture 2:

```
OPEN 1,1,1,"FICH"
PRINT 1, A1
PRINT 1, B1
CLOSE 1
```

Lecture 2:

```
OPEN 1,1,0,"FICH"
INPUT 1,A1
INPUT 1,B1
PRINT A1
PRINT B1
```

La première solution me paraissait moins compliquée ou alors, est-ce que j'ai fait une erreur ? Mon VIC en est bouillant de fièvre. Au fait, combien de temps peut-on laisser allumé le VIC, vu les calories dissipées par la régulation ?

CONCLUSION :
Et dire qu'il y a 20 ou 30 ans, les veillées au coin du feu ... . Mais, en ces temps un peu tristes, il est encore permis de se payer une pinte de bon sang. Hi ! Hi ! Le rire étant et restant le propre de l'homme ET VIVE L'INFORMATIQUE.

Si ces (trop longues) lignes peuvent aider "un plus débutant que moi" -cf. La Commode n°5- alors tant mieux.

La Pratique du VIC, vol. 1 est arrivée entretemps. On y trouve

p.21 (c'est p.18 dans "La Pratique du CBM vol.1) des considérations sur la nécessité de la présence sur la bande d'un retour-chariot ou d'une virgule entre chaque variable qui expliquent le problème que vous avez rencontré.

Quant au problème des arrondis, il n'a pas fini de faire couler de l'encre. Il faut être méfiant vis-à-vis des ordinateurs.

Arrondis et autres questions

Ce n'est malheureusement que chez un seul revendeur à STRASBOURG que j'ai pu me procurer votre revue et même des anciens numéros. Bravo pour La Commode; elle est très bien et merci aussi pour les articles de débutant que je suis.

Etant débutant, je me permets de vous poser quelques questions:

1) Envisagez-vous (ou quelqu'un d'autre) la traduction du livre du VIC 20 Programmer's guide ?

2) Allez-vous publier une suite à la "Lettre de La Commode" dont vous m'avez envoyé le n°1 mai 1982 ? Si oui, les abonnés à la revue recevront-ils ces lettres ?

Voici encore quelques questions techniques de mes enfants :

3) Notre télé se trouve dans un meuble fermé, avec deux battants et pas de fond avec une bonne circulation d'air tout autour. A 16 cm audessus de la télé se trouve une étagère dans ce meuble qui serait l'emplacement idéal pour ranger le VIC 20. Y-at-il un inconvénient à ranger l'ordinateur à cet endroit ? Peut-on aussi ranger les cassettes, et plus tard, les disquettes sur cette étagère ? (Les bandes et disques sont-ils influencés par l'alimentation du haut-parleur de la télé ?). Voilà un article à publier : "Comment protéger les bandes et disques" et "entretenir le matériel".

4) Ci-joint un programme. Pourquoi dans le 1er exemple l'ordinateur donne-t-il .999999999 au lieu de 1?

5) Quelle dimension des vecteurs faut-il prendre : DIM ou DIMM ?

6) Pour INPUT : le message doit-il être : /= à 16 lettres et espaces ? (Si plus de 16 lettres ou espaces l'ordinateur ne le prend pas).

7) Pour travailler sérieusement quel est le prochain achat à faire (dans l'ordre d'importance): Extension mémoire 16 K, ou disquette ou imprimante ? (Je possède le VIC 20 avec le cassettophone)

```
10 REM EQUATION DU SECOND DEGRE
20 INPUT "LE COEFFICIENT DU CARRE:"
     ; A
30 IF A = 0  THEN PRINT "CE N'EST
     PAS UNE EQUATION DU SECOND
     DEGRE";
40 INPUT "LE COEFFICIENT DE X:"; B
50 INPUT "LA CONSTANTE:" ; C
60 D = (B2) - 4*A*C
70 IF D   0 THEN GOTO 100
80 IF D = 0 THEN GOTO 138
90 PRINT "IMPOSSIBLE" : GOTO 150
100 P = (-B + SQR(D)) / (2*A):
     S = (-B - SQR(D)) / (2*A)
110 PRINT "PREMIERE RACINE:";S:
     GOTO 150
130 X = - B / (2*A)
140 PRINT "RACINE DOUBLE:";X
150 END
```

Par exemple:
résoudre $X^2 - 3X + 2 = 0$

RUN RETURN
Le coefficient du carré = 1 RETURN
Le coefficient de X = -3 RETURN
La constante : 2 RETURN
Première racine : 2
Deuxième racine : .999999999 au lieu de 1. Approximation ?

Par contre, pour résoudre: X2-1 = 0
RUN RETURN
Le coefficient du carré : 1
Le coefficient de X : 0
La constante : - 1
Première racine : 1
Deuxième racine : - 1 (ce qui est juste)

Pourquoi l'ordinateur n'affiche-t-il pas aussi 1 au lieu de .999999999 dans la 1ère équation ?

La racine double marche et le D [0 impossible aussi.

Richard HAAS
67150 ERSTEIN

Voici quelques réponses à vos questions :

1) Le livre VIC 20 Programmer's reference guide est en cours de traduction, en partie par notre rédacteur en chef, D. TRECOURT. Dès que le livre sera en vente, La Commode l'annoncera, bien sûr.
Sinon, les livres de D.J. DAVID déjà parus sont (chez P.S.I):

-La Découverte du VIC, vol. 1
-La Pratique du VIC, vol. 1 (périphériques et fichiers)
-Clefs pour le VIC

A paraître (ordre chronologique):

-La Pratique, vol. 2 (langage machine)
-La Découverte, vol. 2 (exercices et problèmes)

2) La Lettre de La Commode est un simple document publicitaire.

3) Il n'y a aucun inconvénient à faire les rangements que vous prévoyez.

4) .999999999 est une approximation de 1 à 10^{-9} près, ce qui est la précision de calcul du VIC.

5) DIM s'écrit avec un seul M.

6) Si le message d'interrogation du INPUT dépasse la ligne, le mieux est de le suivre d'un curseur bas.

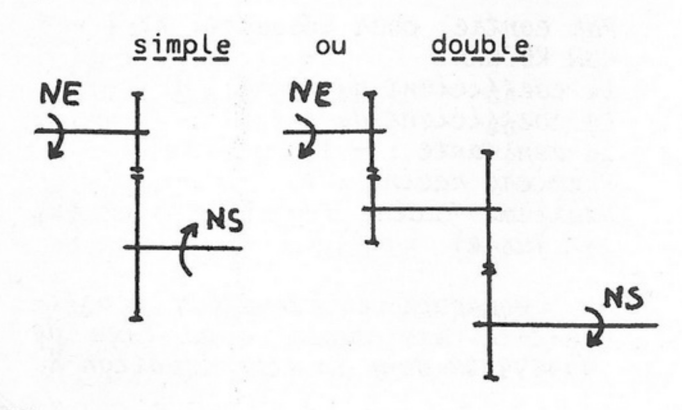
7) Les trois extensions que vous proposez sont aussi essentielles les unes que les autres. Si je devais faire des achats échelonnés, je suivrais l'ordre : disquette, imprimante, extension mémoire, mais cela dépend des applications.

*
* *

Bourse d'échange des programmes

Calcul de réducteurs à engrenages

Ce programme a été développé sur un CBM 3032, et nécessite au minimum 8K de RAM; mais impérativement les ROM'S BASIC 3.0 à cause d'un Hardcopy en Assembleur. Par ailleurs j'ai développé une version sans imprimante, convenant au BASIC 4.0. Le programme nécessite également l'emploi de la Rom EDEX. Le programme optimise un train d'engrenage en le considérant comme:



suivant des considérations venant de l'optimisation dimensionnelle due au rapport de réduction demandé.

* Les données

Puissance à transmettre en W
Vitesse d'entrée du réducteur NE en tr/min
Vitesse de sortie désirée NS en tr/min
Coefficient de service dépendant du milieu, des contraintes de travail
Type de construction
Matériau choisi parmi le catalogue

* Résultats fournis

M module normalisé
Z1, Z2 nombres de dents
D1, D2 diamètres primitifs
Entraxe
Largeur de denture
Effort tangentiel
Couple transmis

Prix TTC jusqu'au 31-1-84
Sauf erreur ou modifications tardives ou épuisement stocks.

CHEZ DURIEZ :

- Après-vente, garantie un an : le 1er mois, échange ; ensuite prêt sous caution.
- Duriez est ouvert de 9 h 30 à 19 h., du Mardi au Samedi, 132, Bd Saint-Germain, 6e, M° Odéon.

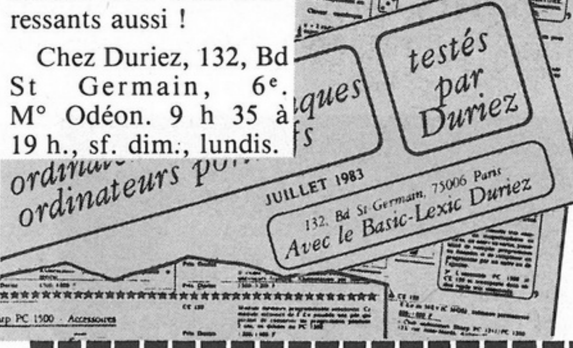
Quand Duriez critique Sharp... (etc.)

IL EST SORTI ! Le nouveau catalogue Duriez "micro-ordinateurs" considérablement augmenté, contient plusieurs pages sur Sharp, Commodore, Atari, Sinclair, Oric, Casio, Epson, Thomson, Sanyo,...

Il est bourré d'appréciations critiques, tests, opinions d'utilisateurs. Plus le Lexic-Basic Duriez. Envoi contre 3 timbres à 2 Fr. (Gratuit au magasin).

Et les prix charter Duriez sont bien intéressants aussi !

Chez Duriez, 132, Bd St Germain, 6e. M° Odéon. 9 h 35 à 19 h., sf. dim., lundis.



Banc d'essai Duriez des 20 micro-Ordinateurs de pointe

24 pages. Envoi contre 3 timbres. Utilisez le Bon ci-dessous (gratuit au magasin).



Avez-vous vu les prix Duriez ?

SINCLAIR ZX 81

ZX 81	580
Mémoire 16 K	360
Imprimante	690
Spectrum 16K Péritel	1850
Spectrum 48K Péritel	2325

THOMSON TO.7

MATERIELS

Unité centrale	2480
lecteur enregistreur	690
Extension 16 K	750
Contrôleur COM.	850
Son + manette de jeux	580
Control. + lecteur disq.	3800
Lect. disqu.	2600
Memo basic	480
Imprimante impact	2500
Cordon imprimante Thermique	250
Cordon imprimante impact	350
T.O. 7 + Mémo-Basic + sonmanette + logiciel Pictor et Trap.	3490
Mots croisés vol. I	195
Mots croisés vol. II	195
Cocktail vol. I	95
Basic vol. II	195
Mots en fleurs	195
Ronde des chiffres	125
Carte de France	145
Noix de Coco	145
Bibliothèque	490

LOGICIELS

Atomium cartouche	350
Echo cartouche	260
Surveyor cartouche	350
Logicod cartouche	295
Gemini cartouche	260
Crypto cartouche	295
Motus cartouche	295
Tridi cartouche	260
Trap cartouche	375
Pictor cartouche	495
Melodia cartouche	495
Sauterelle cassette	125
Basic vol. I	195
Comp. et Mult.	120
Syst. métriq. cassette	145
Carré magique cassette	175
L'Horloge cassette	125
Encadrement cassette	120
Carotte cassette	175

Ordinateur HP75C

HP 75 C	8190
Module mémoire 8 K	2103
Module Math I	480
Module Math II	480
Module Math III	480
Module Stat	480
Module électronic	480
Module Finance	480
Module Test Stat	480
Module Game I	480
Module Game II	480
30 cartes magnétiques	360

PERIPHERIQUE HP IL

Module HPIL	1348
Cassette digital	3950
Imprimante thermique IL	3950
Interface moniteur	2465
Interface TV 3350	3350
Mini cassettes (10)	1138

ORIC-1

Version 48 Ko avec Peritel	2120
Magnéto	380
Traceur 4 coul.	1900

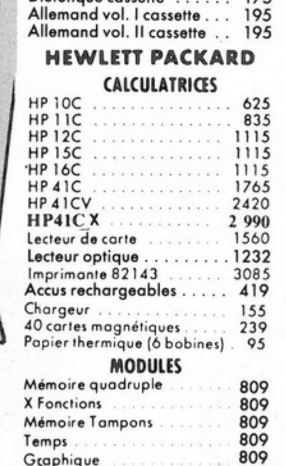
COMMODORE VIC 20

Vic 20 micro ordinateur 3,5 K avec NB (UHF)	1590
Vic 20 Secam	2270

PERIPHERIQUES

Vic 1530 lecteur enregistreur de cassette	370
Vic 1541 unité de mono disquette 170 K	3160

400 micro-prix-charter en direct chez Duriez



Diététique cassette	175
Allemand vol. I cassette	195
Allemand vol. II cassette	195

HEWLETT PACKARD

CALCULATRICES

HP 10C	625
HP 11C	835
HP 12C	1115
HP 15C	1115
HP 16C	1115
HP 41C	1765
HP 41CV	2420
HP41CX	2 990
Lecteur de carte	1560
Lecteur optique	1232
Imprimante 82143	3085
Accus rechargeables	419
Chargeur	155
40 cartes magnétiques	239
Papier thermique (6 bobines)	95

MODULES

Mémoire quadruple	809
X Fonctions	809
Mémoire Tampons	809
Temps	809
Graphique	809

GP 100 VC imprimante 80 col 30 cps	2420
Cordon Péritel	165

EXTENSIONS

Vic 1020 coffret extensions	1 350
Vic 1210 cartouche extension 16K	665
Vic 1110 cartouche extension 8K	395
Vic 1011 A terminal RS232C	320
Vic 1311 manche à balai	130
Vic 1312 manette de commande (paddle)	177

AIDE A LA PROGRAMMATION

Vic 1211M cartouche super extender	426

PROGRAMMES EDUCATIFS ET SCIENTIFIQUES

Autoformation au Basic (cassette)	415
Bibliothèque MATH STAI (disquette)	533
Vic GRAF (cartouche)	379
Vic STAI (cartouche)	379
Vic FORTH (cartouche)	581
Vic RELAY (cartouche)	462
Vic 3302 Simplicalc (cassette)	420
Vic 3301 Simplicalc (disquette)	490
Vic Stock (cassette)	420
Vic Stock (disquette)	490
Vic 3306 Vic writer (cassette)	490
Vic 3305 Vic writer (disquette)	490
Vic 3304 Vic File (disquette)	490

PROGRAMMES RECREATIFS

Vic 1901 Avengers (cart)	213
Vic 1902 Star Battle (cart)	213
Vic 1904 Super slot (cart)	213
Vic 1906 Alien (cartouche)	213
Vic 1907 Jupiter Lander (cart)	213
Vic 1908 Poker (cart)	213
Vic 1909 Road race (cart)	213
Vic 1919 Sargon 2 Chess	266
Vic 1910 Rat race (cart)	213
Vic 1914 Adventureland (cart)	270
Vic 1515 Pirate cove (cart)	270
Vic 1916 Miss. imposs. (cart	270
Vic 1917 The count (cart)	270
Vic 1918 Voodoo castle (car)	270
Vic 1912 Mole attak (cart)	213
Vic 3501 Quizmaster (cart)	190

COMMODORE 64

Commodore 64 PAL	2790
64 SECAM	3650
Lecteur enregistreur de cassette = VIC 1530	370
Unité de mono disquette 170K = VIC 1541	3160
GP 100 VC imprimante 80 col 30 cps	2420
TOOL 64 utilitaire (cartouche)	640
FORTH 64 (cartouche)	690
CALCRESULT (disquette)	2.312
STAT 64 (disquette)	490



aux prix Duriez

SANYO PHC 25

PHC 25	1.790
Cordon Peritel	108
Cordon magneto	65
Cordon Imprimante	280

SHARP

PC 1212	745
Imprimante CE 122	850
PC 1500	1750
Imprimante CE 150	1770
PC 1500 + CE 150	3400
Extension 8K CE 155	450
Extension 16K CE 161	1700
Extension 8K protégeable CE 159	1100
Interface RS232C parallèle	1890
Cable imprimante	580
Clavier sensitif	1.240
PC1251	1.390
Imprimante CE 125	1.590
PC 1251 + CE 125	2900
PC 1245 + CE 125	2500
PC 1245	780
PC 1245 + CE 125	2300
MZ 720	2970
Traceur 4 coul.	1750

CASIO

Fx 702P	1.050
Interface magneto FA 2	260
Imprimante FP 10	560
FX 802 P	1400
PB 100	645
Interface magneto FA 3	245
Imprimante FP 12	635
FP 200	2990
Extension 8 Ko	623
Câble K7	94
Secteur	222
Traceur 4 couleurs	2470
Unité de disque 70 Ko	4732
Câble imprimante	405
PB 700	1662
Traceur 4 coul.	2280
Magneto	850
Mémoire 4 K°	427

CANON

X07 8 Ko	2170
Mémoire 8 Ko	780
Carte mémoire permanente 4 Ko	389
Imprimante-tracer 4 coul.	1650
Câble magnéto	49
Coupleur optique	420
X07 traceur	3700

EPSON

HX 20	5960
Lecteur cassettes	1280
Extension 16 Ko	1170
Modem	1580

IMPRIMANTES

Seikosha GP 100 A	2190
Seikosha GP 250	3190
Seikosha GP 700	4850
Brother EP 22	2500
Brother CE 60	5050
Interface IF 50	2315
Epson RX 80	3884

Je commande à Duriez : Duriez, 132, Bd St-Germain, 75006 Paris.

☐ 1 Catalogue Duriez "Micros" (essais comparatifs des 20 micro-ordinateurs les plus vendus chez Duriez) contre 3 timbres à 2 F.

☐ Le(s) article(s) entouré(s) sur cette page photocopiée (ou cités ci-dessous).

Si changement de prix, je serai avisé avant expédition.

☐ Ci-joint chèque de ....
.... F
y compris Port et Emballage 40 F

☐ Je paierai à réception (Contre Remboursement) moyennant un supplément de 30 F + 40 F Port et emballage.

J'aurai le droit, si non satisfait, de renvoyer sous 8 jours le(s) appareil(s) modules, Cassettes ou ouvrages Duriez, qui me remboursera la somme ci-dessus, (sauf suppl. 30 F du C. Rb), port et emballage.

Mes Nom, Prénoms, Adresse (N°, Rue, Code, Ville) :
....
....
....

Date et Signature ....
....

Media Conseil

MAGAZINE

Une GP 100 pour l'Atari

Une version de la GP 100 adaptée par AXIOM sera commercialisée sous le nom de AT 100. Elle se connectera à tous les modèles d'ATARI, 400, 800, 600, etc., directement, sans nécessiter le module d'interface 850.

Au feu! Au feu!

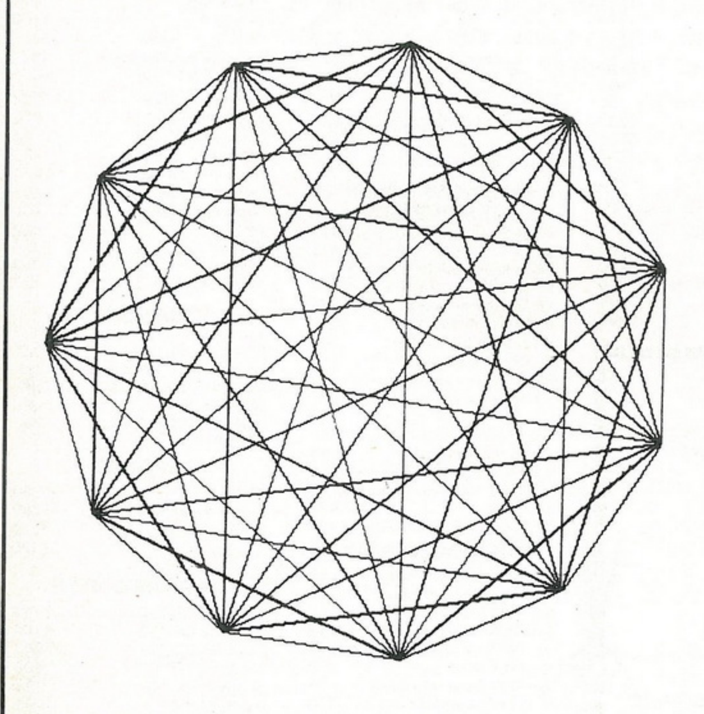
Un stock de 25000 ORIC a brûlé dans un incendie aux usines d'ORIC en Angleterre. 25000 machines, c'est le quart de la production d'ORIC de l'année dernière. L'importance du sinistre a évidemment déclenché les mauvaises langues qui parlent de prime d'assurance. D'autres mauvaises langues -encore plus mauvaises- demandent si les ORIC qui ont brûlé étaient en panne (?)

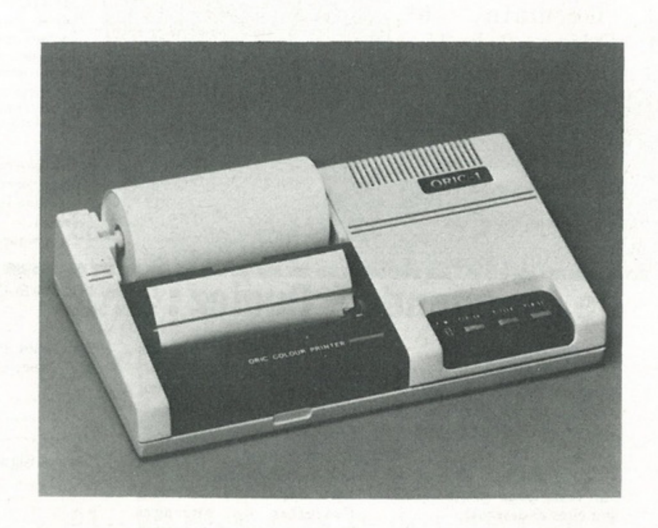
Imprimante couleur pour l'ORIC et les VIC/C64

Il s'agit de versions personnalisées de l'imprimante-traceuse BFM 10 qu'on pouvait voir à la Micro Expo de juin dernier. La version COMMODORE (photo dans le n° 8) reçoit le jeu de caractères COMMODORE, l'interface IEEE série et le nom VIC 1520.

La version ORIC (photo ci-dessous) reçoit l'interface parallèle Centronics. Noter qu'elle pose les mêmes problèmes de timing vis-à-vis des interruptions que l'imprimante SEIKOSHA (Voir ORIC à BRAC de ce numéro).

Il s'agit, en fait, d'un véritable traceur de courbes capable de choisir entre 4 stylos; voir Ex. ci-contre que nous reproduisons malheureusement en noir et blanc.





Fin d'un concurrent du VIC

La presse non spécialisée a consacré une place importante au fait que TEXAS cesse de produire son micro-ordinateur domestique TI 99/4A après avoir perdu plus de 100 millions de dollars en un trimestre. Il s'agit, bien sûr, d'une décision américaine car, en France, le TI 99 ne marchait pas mal: il était même le plus dangereux concurrent du VIC.

Cette cessation de production n'est pas sans susciter des inquiétudes pour d'autres marques, car elle manifeste que l'informatique domestique n'a pas pris aussi vite que ses promoteurs ne l'espéraient. Pour notre part, nous aurions tenance à porter ce fait au débit de l'absence de logiciels adéquats donnant à l'ordinateur domestique une véritable utilité.

Parmi les inquiets, ATARI, dont les modèles de la nouvelle gamme tardent, tardent... Le fait d'être soutenu par un groupe géant (la WARNER) est un atout, mais à double tranchant : c'est souvent les grands groupes qui se séparent le plus vite des divisions non bénéficiaires. Or ATARI, tout en restant le premier du côté des consoles de jeu, a commencé à subir les contrecoups du rétrécissement de ce marché et de l'apparition simultanée de plusieurs concurrents.

Les gagnants

APPLE lui-même annonce des pertes, de sorte que les deux seules compagnies à avoir tiré leur épingle du jeu en micro cette année semble être IBM, qui vend plus son nom (et une réputation de service irréprochable) qu'une machine spécialement performante (l'IBM PC est correct mais moyen sur tous les plans) dans le domaine professionnel, et COMMODORE avec le succès éclatant du 64 dans le domaine personnel.

Mais les distributeurs ne doivent pas s'y tromper : le marché devient de plus en plus difficile et, plus que jamais, la compétence, le service rendu à la clientèle, le choix des logiciels seront déterminants. Au fait, un bon critère de choix de votre distributeur : qu'il ait LA COMMODE en magasin; ça prouvera toujours qu'il se tient au courant de ce qui existe !

Radio-micro

Beaucoup de radios - surtout les radios libres - s'intéressent à la Micro-informatique. Voyez dans notre page "Communiqués de presse" une radio de l'ouest.

Un lecteur, Ch. LAPORTE, nous en signale une dans le nord : RBL (Radio-Boulogne Littoral) sur FM 103,7 MHz (BP 162, 62203 BOULOGNE SUR MER CEDEX) a une émission sur la micro intitulée "GOTO 103,7" chaque dimanche de 20h à 21 h.

En région parisienne, il y en avait trois, mais l'une d'elles a été saisie (Radio Paris Ile de France). Reste :

- Radio Rivage Contact (1, av.Pasteur 93140 BONDY - tél.850.17.74) sur 92,45 à 92,1 MHz avec l'émission "écran magique" chaque samedi de 15h30 à 17h,

- Radio Ici et Maintenant (49,rue des Passonniers 75018 Paris,tél.223.42.89) sur 96,6 MHz. qui parle de micro chaque samedi de 14h à 16h.

Nous vous tiendrons au courant d'expériences de transmission de programmes que nous tenterons avec ces radios.

Une bonne Télé

Un lecteur, M. FABRE, de Carmaux, nous signale que les téléviseurs "Philips" de la famille TVC 12 disposent d'une entrée video mixte qui traite, indifféremment, des signaux couleurs, qu'ils soient PAL ou SECAM.

Sur une TV type 18C1205 (46 cm) les couleurs d'un VIC 20 ou d'un 64 sont magnifiques et le son excellent. Il faut activer la prise pé-

ritel par l'inverseur TV-AV, ou en envoyant du +12V sur le contact 8 de la prise et la masse en 5.
Raccordements :
Vidéo : entrée contact 20, masse vidéo 18.
Son : entrée sur 2 ou 6 (indifférent), masse sur 4.

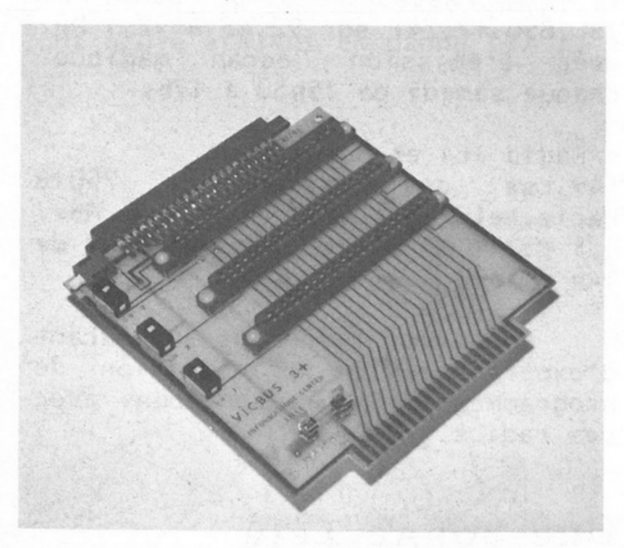
Extensions dans le Nord

Vous connaissez les constructions de notre lecteur, Ch.THERY. Eh bien, elles sont commercialisées par INFORMATIQUE CENTER à Lille. Ce sont :

- Carte 8250-8050 qui permet d'utiliser une 8250 au choix comme 8250 ou 8050. Prix: 648 F TTC (auquel il faut ajouter le prix d'un jeu de ROM 8050)

- Carte MULTIROM pour CBM qui permet, selon la version, de commuter entre deux jeux de 5 ROM (Ex. BASIC 2.0 à BASIC 4.0) ou 5 jeux de 2 ROM
Prix : 884 F TTC.

- VIC-BUS 3+, fond de panier avec 4 connecteurs pour VIC. (490 F TTC):



Cette carte est un concurrent direct de l'EXT-VIC-BUS 4-S de SEDERMI à ceci près que dans la carte SEDERMI, les interrupteurs ne font que désactiver la sélection d'adresse A000, tandis que dans VIC-BUS 3+ ils coupent l'alimentation. Il se trouve que, selon les applications, c'est plutôt l'une ou plutôt l'autre version qui est préférable; il est donc heureux qu'on trouve les deux sur le marché.

- Interface son et télé N/B sur 64.

Ce système s'installe dans un 64 et donne le son meilleur que par la télé et le noir et blanc. Nous aurions préféré un système qui ne donne que le son, car il serait beaucoup plus facile à installer et, pour la télé, l'utilisateur a déjà une solution puisqu'il a le 64. Prix: 190 F TTC.

LORICIELS

Pour ORIC et bientôt pour 64. Cette société qui vient de se constituer (17,rue Lamandé 75017 Paris) a déjà disponibles 25 logiciels pour ORIC dont les derniers sortis sont, entre autres :

Godilloric 95 F

A l'aide de votre "godillot", vous devez écraser sans pitié une colonie de fourmis qui tente de dévorer votre sucre... dépêchez-vous car la chaussure monte et les scorpions arrivent !

Le Protector : 95 F

Au premier tableau, vous survolez une ville dont vous avez la responsabilité. Hélas, vous n'êtes pas seul, les soucoupes ennemies passent et repassent en larguant des envahisseurs. Au second tableau, vous devez faire face à une pluie de météorites. Avec un peu de chance, vous apercevrez le troisième tableau où les dernières forces ennemies sont à détruire.
La rapidité de ce jeu est modulable de 1 à 10.

Orion : 95 F

Mouches robotisées, base spatiale, vaisseau d'attaque... tout y est ! avec graphisme, couleur et rapidité.

Gastronon : 95 F

Un jeu complètement fou : il faut détruire avec une base spatiale, des choucroutes, des cornichons, des chopes de bière à moins que vous ne préfériez le morceau de gruyère suivi d'une petite glace !

J'apprends l'anglais : 140 F

Idéal pour apprendre son vocabulaire. Au fur et à mesure, constituez votre fichier de mots anglais, avec leur traduction. A tout moment, il est possible de faire des recherches, d'éditer des listes...sur écran ou sur imprimante.

Mais ce n'est pas tout, un système d'interrogation permet de faire le point...et attention, l'ordinateur connaît vos points faibles ! Livré avec manuel et, en prime, le fichier des verbes irréguliers.

Annuaire : 140 F

Nom, prénom, adresse, téléphone, profession...un vrai fichier ! Allant jusqu'à l'édition sur imprimante, d'étiquettes autocollantes pour votre courrier. Chaque fichier ouvert possède une capacité de plus de 170 fiches. Détail technique: c'est un fichier multicritère: rien de plus facile que de rechercher toutes les fiches de vos amis habitant Paris, de prénom Jacques et nés en 52...

Facile d'emploi. Livré avec manuel.

Gestion de stock : 180 F :

Logiciel de type professionnel, très complet. Par fichier, vous aurez une capacité de plus de 400 fiches article et quelques 50 fiches fournisseur.

Chaque fiche article contient référence, mnémonique, désignation, emplacement magasin, prix, quantité en stock, quantité minimale, la date du dernier mouvement,...

Idéal pour un commerçant ou tout simplement pour vous qui désirez gérer votre congélateur, votre cave, ou votre grenier... Possibilité de mouvement sur fiche (entrée/sortie), de recherche multicritère, édition d'inventaire ...

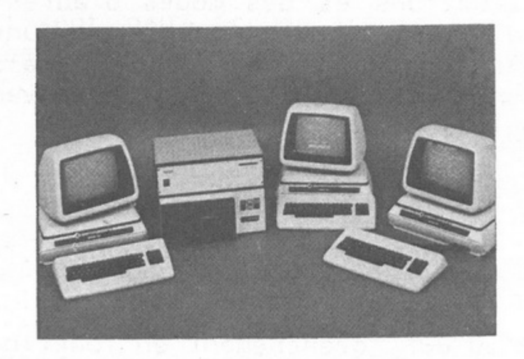
Livré avec manuel.

Enfin, le jeu de KIKEKANKOI fait l'objet d'un concours où vous pouvez gagner un magnétoscope. Renseignez-vous.

Commodore nouveautés haut de gamme

Les livraisons des CBM 700 commencent. Au niveau 500, plus rien ne filtre.

Sinon, le CBM 8000 SK poursuit sa carrière. Il peut devenir multiposte dans le cadre du système MULTI 4 qui permet de connecter jusqu'à quatre 8000 à un même disque dur 20 Mo (10 fixes, 10 amovibles) P8140 sous MEM/DOS 6502. Toutes les publicités de ce système se font sur APPLE, mais rappelons ici qu'il est accessible aux CBM.



Du côté des imprimantes, c'est maintenant deux imprimantes à marguerite qu'on trouve dans la gamme COMMODORE: la 8026B (à clavier) et la 8028 (sans clavier mais 40 c/s). Toutes deux peuvent recevoir l'introducteur de feuilles CBM 8025.



Une imprimante à 700 F pour VIC et ORIC

C'est l'imprimante ZX ! Il existe en Angleterre une interface qui permet de l'utiliser avec le VIC et qui fonctionne très bien.

Nous l'avons essayée : on obtient même les caractères graphiques. Qui l'importera en France ?

Maintenant, la manip. ne devrait pas être difficile à étendre à l'ORIC. Nous en reparlerons.

65 C 02

C'est la version CMOS (donc très faible consommation) du 6502 qu'on attendait depuis longtemps. Elle est produite par ROCKWELL. Mais il y a mieux : le 65C02 a des instructions et des modes d'adressage en plus. Nous publierons une étude complète là-dessus, mais disons maintenant les principales innovations :

Instructions nouvelles :

BRA, branchement inconditionnel
STZ, mise à zéro d'une case mémoire
BBR ou BBS, branchement en fonction de l'état d'un bit déterminé d'une case mémoire
RMB ou SMB mise à 0 ou à 1 d'un bit déterminé d'une case mémoire
TRB ou TSB test d'un bit avec remise à 0 ou à 1.
Empilement et dépilement de X et Y.

Nouveaux modes d'adressage :

indirect page zéro sans index

Modes nouveaux pour des instructions existantes :

JMP reçoit le mode indirect , X
INC et DEC reçoivent le mode accumulateur.

Il va sans dire qu'un certain nombre de codes non spécifiés sont maintenant utilisés et que ceux qui les utilisaient pour les opérations spéciales qu'ils semblaient produire avaient tort.

Rachat or not rachat ?

Depuis le temps que les bruits courent sur le rachat de PROCEP par COMMODORE, rien n'est encore fait. En fait, COMMODORE est en train de négocier avec le gouvernement français l'autorisation de procéder à une simple <u>prise de participation</u> dans PROCEP. A quel pourcentage, nous ne le savons pas.
Au fait, pourquoi y-a-t'il besoin d'une autorisation pour une simple prise de participation? Ne serait-il pas question aussi d'une usine de montage en France comme cela avait déjà été négocié, sans aboutir, sous VGE ?

Quelques logiciels pour VIC (et ATARI)

Un certain nombre de logiciels intéressants nous ont été communiqués. Nous publierons le banc d'essai complet de certains d'entre eux dans les prochains numéros. En attendant, quelques indications :

ANIGRAPHE de ALCOM (VIC+8K) : 250 F

18 commandes BASIC supplémentaires
animation simultanée de 32 dessins haute résolution (analogues aux sprites)
écran de 32 lignes de 23 caractères
scrolling multiple, etc ...

Les autres viennent d'Angleterre et nous ont été communiqués par SIDEG :

HOME OFFICE de NAVAJO SOFTWARE : 285 F

Formé d'un mini-traitement de textes d'une part et d'autre part d'une base de données sur disque ou cassette.

HOME INVENTORY de AUDIOGENIC: 285 F

Permet de dresser l'inventaire de votre patrimoine pour en suivre la valeur ou mieux l'assurer, catégorie par catégorie. Ex. Objets d'art, immeubles ou même les bouteilles de votre cave.

CAR COSTS de AUDIOGENIC: 250 F

Permet de gérer le gouffre que constituent vos dépenses pour votre voiture.

HOUSEHOLD FINANCE de AUDIOGENIC : 285 F

Finances familiales : revenus et dépenses, répartition des dépenses par catégories, budget prévisionnel, frais déductibles des impôts. Stocke les données sur disque ou cassette.

N.B : Les quatre programmes de AUDIOGENIC existent en version ATARI et en version VIC (et aussi 64, probablement).

Adresses des Routines Arithmétiques sur Commodore Atari et Oric

Les routines arithmétiques utilisent deux accumulateurs ACC1 et ACC2. Toutes les adresses sont en hexa. Le Basic de l'Atari n'étant pas Microsoft, les détails des opérations sont légèrement différents.

NOM	OPERATION	3000	4/8000	VIC	64	ORIC	ATARI
ACC1	1er accumulateur	5E-63	5E-63	61-66	61-66	D0-D5	D4-D9
ACC2	2nd accumulateur	66-6B	66-6B	69-6E	69-6E	D8-DD	E0-E5
MACC1	ACC1 ←-- Mem(Y,A)	DAAE	CCD8	DAD3+	BBA2	DE73	DD8D++
MACC2	ACC2 ←-- Mem(Y,A)	D998	CBC2	DA8C	BA8C	DD4D	DD98
ACC1M	Mem(X,Y) ←-- ACC1	DAE0	CD0A	DBD4	BBD4	DEA5	DDA7
ACC21	ACC1 ←-- ACC2	DB08	CD32	DBFC	BBFC	DECD	
ACC12	ACC2 ←-- ACC1	DB18	CD42	DC0C	BC0C	DEDD	DDB6
ZERO	ACC1 ←-- 0	D803	CA2D	D8F7	B8F7	DB27	DA44
OPPOS	ACC1 ←-- ACC1	D853	CA7D	D947	B947	E26Dx	
ARRON	arrondit ACC1	DB27	CD51	DC1B	BC1B	DEEC	
NORMAL	normalise ACC1	D7E3	CA0D	D8D7	B8D7	DB07	
ADD	ACC1 ←-- ACC2+ACC1	D776	C9A0	D86A	B86A	DA9A	DAC6
ADDM	ACC1 ←-- Mem +ACC1	D773	C99D	D867	B867	DA97	
SUB	ACC1 ←-- ACC2-ACC1	D736	C989	D853	B853	DA83	DA60**
SUBM	ACC1 ←-- Mem -ACC1	D733	C986	D850	B850	DA80	
MULT	ACC1 ←-- ACC2*ACC1	D93C	CB61	DA30	BA30	DCBA	DADB
MULTM	ACC1 ←-- Mem *ACC1	D934	CB5E	DA2D	BA2D	DCB7	
DIV	ACC1 ←-- ACC2/ACC1	DA1E	CC48	DB12	BB12	DDE3	DB28**
DIVM	ACC1 ←-- Mem /ACC1	DA1B	CC45	DB0F	BB0F	DDE0	
FPEXP	ACC1 ←-- ACC2↑ACC1	DE68	D112	DF7B	BF7B	E231*	
ABS	ACC1 ←-- ABS(ACC1)	DB64	CD8E	DC58	BC58	DF31	
ATN	ACC1 ←-- ATN(ACC1)	E08C	D32C	E30B	E30E	E43B	
COS	COS	DFD8	D282	E261	E264	E387	
EXP	EXP	DEDA	D184	DFED	BFED	E2A6	DDC0
INT	INT	DBD8	CE02	DCCC	BCCC	DFA5	
LOG	LOG	D8F6	CB20	D9EA	B9EA	DC79	DECD
SIN	SIN	DFDF	D289	E268	E26B	E38E	
SQR	SQR	DE5E	D108	DF71	BF71	E22A	
TAN	TAN	E028	D2D2	E2B1	E2B4	E307	
INTFLP	(A.Y) --→ flottant	D26D	C4BC	D391	B391	D3ED	D9AA
FLPINT	ACC1 --→ entier	D09A	C2EA	D1BF	B1BF	D871	D9D2
FLPASC	convertit ACC1 en ASCII ds $0100 sqq	DCE9	CF93	DDDD	BDDD	E0D1	D8E6

Mem sans précision signifie Mem(Y,A) (= pointée par (Y.A))

* ACC1 ←— ACC2↑Mem(Y,A)

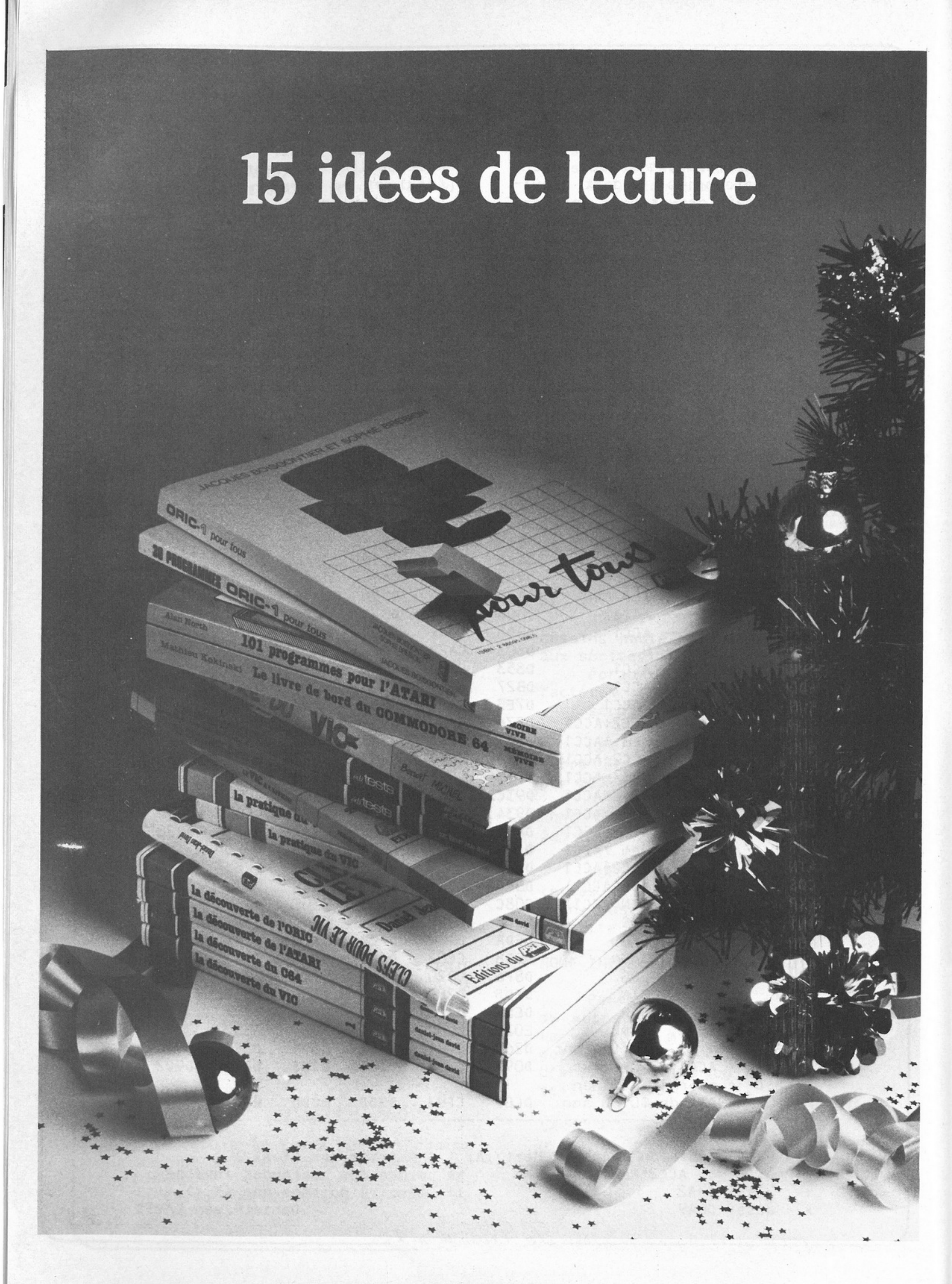
\+ aussi DBA2

x aussi DBA9

** ordre des opérandes inversé

++ mémoire pointée par (X,Y)

Daniel-Jean DAVID


15 idées de lecture
ORIC-1 pour tous
30 PROGRAMMES ORIC-1 pour tous
pour tous
Alan North
101 programmes pour l'ATARI
Mathieu Kokinski
Le livre de bord du COMMODORE 64
MÉMOIRE VIVE
MÉMOIRE VIVE
la pratique du VIC
la découverte de l'ORIC
la découverte de l'ATARI
la découverte du C64
la découverte du VIC
CLEFS POUR LE VIC
Editions du
daniel-jean david
daniel-jean david

Mettez des idées dans votre ordinateur.

La découverte du Vic
par Daniel-Jean David - 176 pages

90,00 FF

Ce livre qui comprend essentiellement une présentation progressive du langage Basic, fait découvrir les divers champs d'application de l'ordinateur individuel Vic. Il aborde spécialement ses points forts : graphiques, sons, couleurs.

Le Vic à l'affiche
par Jean-François Sehan - 144 pages

90,00 FF

Possesseurs de Vic, ce livre s'adresse à vous. Du papier peint à la cuisine, en passant par le Baccara et le Taquin, il regroupe 20 programmes à caractère familial, utiles ou ludiques. Dans un but pédagogique, chaque programme est accompagné d'un organigramme, d'une liste de variables et d'une application de chaque ligne Basic ce qui vous permettra, éventuellement, l'adaptation à d'autres Petits Systèmes Individuels.

Clefs pour le Vic
de Daniel-Jean David - 120 pages

90,00 FF

" Clefs pour le VIC ", édité dans la collection " Mémentos ", permet aux utilisateurs de l'ordinateur familial Vic 20 d'accéder rapidement à une information de référence comme la syntaxe des commandes, les codes caractères, les adresses, les messages d'erreurs etc... Il évite ainsi d'aller rechercher les renseignements nécessaires dans différents manuels. Il offre également un recueil de 40 " trucs " utiles. Destiné à se trouver en permanence à côté de l'ordinateur, il s'agit bien d'un OUTIL indispensable à tout programmeur.

La pratique du Vic
de Daniel-Jean David - 176 pages

90,00 FF

Faisant suite à " la découverte du VIC ", cet ouvrage concerne les périphériques de l'ordinateur familial Vic 20 et leurs applications, de la cassette au disque souple en passant par l'imprimante et le crayon lumineux. Il est complété par de nombreux exemples et exercices avec leur solution.

Le livre du Vic
par Benoît Michel - 248 pages

110,00 F

A la fois guide pour le débutant et ouvrage de référence pour l'amateur averti et le professionnel, le livre du Vic détaille tout d'abord le fonctionnement du Vic 20 grâce à l'étude des éléments du système et des programmes internes du Vic. Sont ensuite abordées les extensions matérielles et logicielles (extensions mémoires, moniteur langage machine). Enfin un index détaillé ainsi que la carte mémoire font du livre du Vic un ouvrage très complet.

La pratique du Commodore 64
par Daniel-Jean David - 176 pages

90,00 FF

De la cassette au disque souple, de l'imprimante aux poignées de jeux et crayon lumineux, ce livre vous donnera tout ce qu'il faut connaître pour utiliser au mieux les périphériques de votre Commodore 64. Conçu dans l'esprit de " la découverte du Commodore 64 " du même auteur cet ouvrage contient de nombreux programmes écrits tant pour les applications personnelles que professionnelles. Un chapitre est consacré aux notions sur les bases de données et au système d'exploitation disque. La programmation de l'interface RS 232 est décrite.

La découverte du Commodore 64
par Daniel-Jean David - 176 pages

90,00 FF

Sans aucune connaissance préalable vous pouvez aborder cet ouvrage dans lequel sont traitées les applications sérieuses mais aussi ludiques de l'ordinateur individuel Commodore 64.
Une introduction générale sur l'informatique, suivie d'un apprentissage progressif du langage Basic vous amène pas à pas à construire vos propres programmes.
Des notions nouvelles sont introduites graduellement et, grâces à elles, vous apprendrez vite à maîtriser les points forts du Commodore : les graphiques, les sons, les couleurs, la haute résolution et les " sprites ".

Le livre de bord du Commodore 64
par Mathieu Kokinski - 256 pages

120,00 FF

Ce livre est une introduction à l'ordinateur individuel Commodore 64, à son équipement, à ses périphériques, ainsi qu'à son langage Basic. Sont aussi abordés : la programmation simple, les possibilités graphiques et musicales, l'utilisation des fichiers, les programmes de tri etc... L'ouvrage est complété de programmes variés, du jeu le plus simple à un mini traitement de textes, en passant par un petit programme de gestion de fichiers.

La découverte de l'Atari modèles 400 et 800
par Daniel-Jean David - 176 pages

90,00 FF

Ce livre d'initiation, qui ne nécessite aucune connaissance préalable, permet de découvrir, non seulement les applications sérieuses, mais aussi les programmations des jeux sur l'ordinateur individuel Atari (400 ou 800).
Après une introduction générale sur l'informatique, le langage Basic est présenté de façon progressive à travers l'élaboration de programmes. Les notions nouvelles sont introduites graduellement au cours des améliorations successives des programmes. Les points forts de l'Atari (graphiques, sons, couleurs) font l'objet d'une étude particulière.

101 jeux, trucs, astuces et programmes pour l'Atari 400/800/1200
par Alan North - 160 pages

90,00 FF

Débutant, apprenti programmeur ou amateur averti, voici 101 programmes, trucs et astuces pour vous et votre Atari : une introduction nécessaire aux nombreuses techniques de programmation de l'Atari, modèles 400, 800 et 1200. Vous y trouverez tout particulièrement des graphiques en couleur, de nombreux programmes de jeux et une foule de " trucs " pour utiliser au maximum les possibilités de votre Atari.

Oric-1 pour tous
par Jacques Boisgontier et Sophie Brébion - 176 pages

100,00 FF

Consacré à un ordinateur individuel tout nouveau et pourtant déjà très populaire, cet ouvrage d'initiation permet de découvrir les grandes qualités graphiques et sonores de l'Oric-1. Tout d'abord, vous assimilerez les notions fondamentales de la programmation (variables, tests, boucles...), puis vous parviendrez aisément, grâce aux nombreux exemples illustrés et aux programmes commentés, à écrire vos propres programmes.

La découverte de l'Oric-1
par Daniel-Jean David - 176 pages

90,00 FF

Le but de ce livre est de découvrir l'ordinateur individuel Oric-1 et son langage Basic. Cette exploration est progressive grâce à l'élaboration de programmes de difficulté croissante, au cours de laquelle les notions nouvelles sont introduites. On aborde spécialement les points forts de l'Oric-1 : graphiques, sons, couleurs, horloge.

30 programmes Oric-1 pour tous
par Jacques Boisgontier - 136 pages

90,00 FF

Voici un ouvrage dans lequel sont regroupés des programmes commentés et illustrés, permettant d'approfondir ses connaissances en Basic. Des exercices et des jeux mais aussi des programmes d'éducation et de gestion invitent à utiliser l'ordinateur Oric-1 avec beaucoup d'originalité et à en découvrir les fonctions particulières.

Les systèmes à microprocesseurs
Daniel-Jean David - 128 pages

90,00 FF

Ce livre vous initiera aux conditions techniques de la révolution micro-informatique. Les différents circuits intégrés : microprocesseurs, mémoires, boîtiers d'entrées-sorties sont décrits et on montre comment les assembler pour former un système.
Les phases du traitement d'une application et du développement d'un système à microprocesseur sont décrites, notamment du point de vue logiciel (programmation en assembleur) et des choix à effectuer.

Mise en œuvre du BUS IEEE utilisation et réalisations d'appareils
Gérard Bastide et Jean-René Vellas - 152 pages

90,00 FF

Après une description complète du bus, ce livre vous apprendra comment mettre en œuvre toutes les possibilités de votre installation : configurer et interroger un périphérique ; le déclencher ou le réinitialiser, etc.
Vous trouverez la description et les syntaxes sur différents calculateurs de toutes les commandes unilignes ou multilignes, universelles ou adressées et la réponse à toutes sortes de questions : comment connaître du premier coup d'œil les capacités d'un périphérique ? Comment le calculateur identifie-t-il un périphérique en détresse ? Comment mettre en communication des calculateurs de marques différentes ? Etc.



P.S.I.

P.S.I. DIFFUSION
BP 86 - 77402 Lagny-S/Marne Cedex
FRANCE
Téléphone (6) 006.44.35

P.S.I. BENELUX
5, avenue de la Ferme Rose
1180 Bruxelles
BELGIQUE
Téléphone (2) 345.08.50

En SUISSE
P.S.I. Suisse
Route Neuve 1
1700 Fribourg
Tél. : (037) 23.18.28

Au CANADA
SCE Inc. 65, avenue Hillside
Montréal (Westmount)
Québec H3Z1W1
Tél. (514) 935.13.14

Envoyer ce bon accompagné de votre règlement à P.S.I. DIFFUSION ou, pour la Belgique et le Luxembourg à P.S.I. BENELUX ou pour la Suisse à P.S.I. Suisse.

NOM ____________________

PRENOM ____________________

Rue ____________________ n° ______

Ville ____________________

Code postal |_|_|_|_|_|

DESIGNATION	Nbre	PRIX
	TOTAL	

Banc d'essai de la cartouche VIC MASTER-SCREEN

Un produit français bien utile

Présentation

MASTER-SCREEN est un produit original de MICRO-APPLICATION, 147, avenue Paul Doumer 92500 RUEIL MALMAISON, et distribué par le réseau PROCEP.

MASTER-SCREEN se présente comme une extension du BASIC. Il s'agit principalement d'un ensemble de nouvelles instructions qui permettent de simplifier l'écriture des programmes et souvent avec de meilleures performances qu'avec une solution entièrement "BASIC".

Outre les fonctions de gestion d'écran et de saisie d'information, qui sont à sa base, MASTER-SCREEN met à la disposition du programmeur trois instructions de manipulation de caractères, ainsi que des commandes DOS facilitant l'utilisation des disques et fichiers sur disque.

La cartouche placée sur le connecteur est activée à la mise sous tension, aucune commande de lancement n'est donc à exécuter pour cela.

Liste des commandes

Nom	Action

Instructions d'affichage:

TLINE	Trace une ligne horizontale
TCOL	Trace une ligne verticale
CLEAR	Efface l'écran à partir d'une origine que l'on précise sur un nombre de positions -limité à 255- que l'on précise aussi.
OUT	Affiche sur l'écran la chaîne de caractères précisée à l'adresse indiquée.
REV	Inverse le contraste dans la zone précisée de l'écran. Possibilité d'imposer une nouvelle couleur à cette zone.
SCROLL	Permet le déplacement d'une ligne vers le haut ou vers le bas (resp.d'une colonne vers la gauche ou vers la droite) de toutes les lignes (resp.colonnes) d'une fenêtre que l'on précise dans la commande. La fenêtre est un rectangle de taille quelconque sur l'écran : au moins 1 col. x 2 lignes (resp. 2 col. x 1 ligne)
SCREEN	Redéfinit la couleur de l'écran (parmi 16), la couleur du bord (parmi 8) et éventuellement (facultatif) la couleur des caractères (parmi 8).

Instructions de saisie :

DECZ	Permet de définir une zone l'écran, en précisant son origine en une position quelconque de l'écran,ainsi que sa longueur (< 255) et le type de contrôle à

effectuer lors d'une saisie. Dans le cas d'une zone numérique on peut en outre préciser son format (en particulier le nombre de décimales).
On peut définir (domaine de définition du numéro de zone) jusqu'à 128 zones.

REQZ Permet de saisir des informations dans la zone indiquée, avec les contrôles automatiques prévus lors de la définition de la zone en question. Le curseur est placé dans la zone, s'y déplace sans en sortir, jusqu'à l'enfoncement de la touche RETURN. Le contrôle est effectué sur 2 niveaux :
- 1 premier niveau pendant la frappe des caractères : certains sont simplement ignorés.
- 1 deuxième niveau, à la fin de la saisie en fonction du format (par exemple nombre de décimales données excessif) : en cas d'erreur la saisie est reprise au début.

INZ Transfère le contenu de la zone indiquée dans la variable indiquée.
Remarque : la notice précise variable alpha-numérique, cependant nos essais ont marché sans erreur avec des variables numériques lorsque la zone était numérique.

OUTZ Affiche la variable indiquée dans la zone précisée.

CLEARZ Efface le contenu de la zone indiquée. Utile avant la saisie d'une nouvelle information dans une zone donnée (REQZ)

REVZ Inverse le contraste dans la zone indiquée. (Préalablement définie par DECZ)

Instructions de gestion de pages d'écran :

SSAVE Sauvegarde, sous un nom que l'on précise, le contenu intégral de l'écran. On peut en plus (facultatif) sauvegarder la structure de certaines zones définies par DECZ.

SLOAD Permet d'afficher à l'écran une page préalablement sauvegardée par SSAVE (a priori c'est là que ces commandes prennent toute leur dimension, par un autre programe).

Compléments BASIC :

UPLOW Transforme le mode majuscule-minuscule en graphique-majuscule pour la variable alphanumérique précisée.

HUNT Cherche, dans une variable alphanumérique, la position du premier caractère égal au caractère précisé. La position est rangée dans la variable ZO (2 lettres)

CREATST Crée une chaîne de longueur indiquée et constituée du caractère précisé, (répété autant de fois qu'il faut).

Commandes DOS :

8 commandes commençant par sont ajoutées au BASIC pour :

- donner le code d'erreur d'un disque ( )
- afficher le catalogue ($)
- initialiser l'unité de disque (I)
- supprimer les fichiers inutilisables (non fermés par exemple) (V)
- formater une cassette (N)
- renommer un fichier ou un programme (R)
- effacer un fichier ou un programme (S)
- copier un fichier (C)

Répétition de commande :

La flèche gauche permet de répéter la plupart des commandes sans réécrire le nom entier de la

commande : seuls les paramètres sont à préciser.

Ex. : TLINE 22,1,1 -- 22,12,1
équivaut à :
TLINE 22,1,1
TLINE 22,12,1

PROGRAMME ILLUSTRANT QUELQUES POSSIBILITES DE MASTER-SCREEN :

Le programme donné en exemple permet de contrôler sa propre connaissance des tables d'addition. Il est volontairement assez sommaire sur le plan ludique, afin de mettre l'accent sur l'utilisation de MASTER-SCREEN pour une gestion d'écran donnée et pour les contrôles des données en entrée.

```
10 MX=10
20 CLEAR255,1,1:CLEAR251,11,14
25 SCREEN1,3,2:TLINE8,3,7:TLINE5
,16,8:SCREEN1,3,6
30 OUT"TABLES D'ADDITIONS",2,2←"
SCORE",15,8←"BONS",18,2←"FAUX",1
8,9←"MOY/10",18,16
50 DECZ1,5,2,3,N,P,"999"←2,5,8,3
,N,P,"999"←3,5,14,4,N,P,"9999"
60 DECZ4,20,1,4,N,P,"9999"←5,20,
8,4,N,P,"9999"←6,20,16,5,N,P,"98
.88"
65 DECZ7,10,4,15
70 OUT"+",5,6←"=",5,12
100 X=INT(RND(0)*(MX+1)):Y=INT(R
ND(0)*(MX+1))
110 X$=RIGHT$(STR$(X),3):OUTZ1,X
$
115 Y$=RIGHT$(STR$(Y),3):OUTZ2,Y
$
120 REQZ3:INZ3,Z:N=N+1
130 IFZ=X+YTHENB=B+1:A$="BONNE R
EPONSE":SCREEN1,3,5:GOTO135
132 F=F+1:A$="REPONSE FAUSSE":SC
REEN1,3,2
135 FOR I=1 TO 5
136 OUTZ7,A$:GOSUB1000
137 CLEARZ7:GOSUB1000
140 NEXTI:SCREEN1,3,6
150 M=10*B/N
160 OUTZ4,STR$(B)←5,STR$(F)←6,ST
R$(M)
165 POKE198,0:REM VIDE TAMPON CL
AVIER
170 CLEARZ3:GOTO100
1000 FORJ=1TO100:NEXTJ:RETURN
READY.
```

Après avoir effacé l'écran, tracé des lignes rouges (SCREEN 1,3,2 définit l'écran blanc, le bord turquoise et les caractères rouges, SCREEN 1,3,6 le redéfinit avec des caractères bleus), affiché divers libellés à divers endroits de l'écran, on définit 7 zones dans les lignes 50 à 65.
Par exemple, la zone 1, définie par: DECZ1,5,2,3, N,P,"999", est une zone de 3 caractères numériques (que des chiffres) en ligne 5, colonne 2.

La ligne 70 place le signe "+" entre les zones 1 et 2 et le signe "=" entre les zones 2 et 3 sur la ligne 5 : c'est sur la ligne 5 que le "problème" est posé. Les lignes 100 à 115 déterminent et affichent les 2 nombres compris entre 0 et MX (MX est initialisé à 10, mais les formats utilisés permettent de le redéfinir à n'importe quelle valeur entre 1 à 999). OUTZ 1, X$ affiche X$ dans la zone 1.

En ligne 120 REQZ 3 assure l'acquisition d'une donnée dans la fenêtre Z3 en ne l'acceptant que si elle satisfait au format défini par DECZ 3,...; ensuite INZ 3, Z permet de transférer cette valeur dans la variable Z. Les lignes 132 et 135 sanctionnent la réponse au problème puis la boucle en lignes 135 à 140 réalise l'affichage clignotant de cette sanction au milieu de l'écran. Le sous-programme en 1000 définit la fréquence du clignotement des sanctions. Le score est affiché sur la ligne 160. Afin d'éviter des ennuis avec une utilisation désordonnée du clavier, le tampon clavier est vidé en 165 et la zone 3 est effacée avant d'aller poser le problème suivant.

Nos conclusions

MASTER-SCREEN semble particulièrement indiqué pour écrire des programmes et même des applications importantes nécessitant une gestion stricte de l'écran : EAO, gestion.

L'ensemble des commandes fournies est riche est l'efficacité de certaines instructions est évidente: principalement les instructions

de saisie et celles de gestion de pages d'écran.

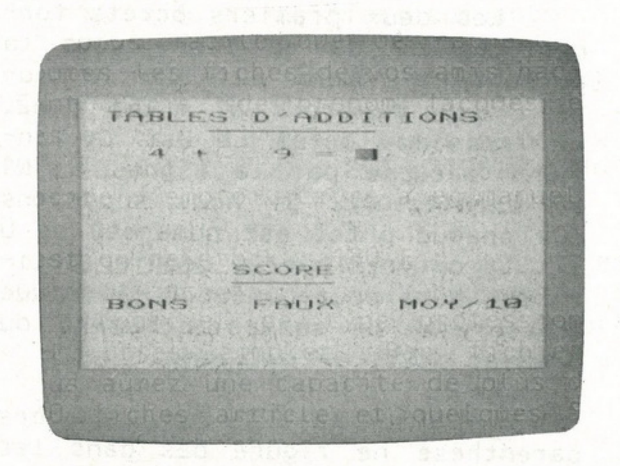
La difficulté résidera dans certains cas (par exemple, écriture d'un titre dans une couleur particulière, éventuellement en contraste inversé...,) dans le choix de la programmation sans ou avec MASTER-SCREEN.

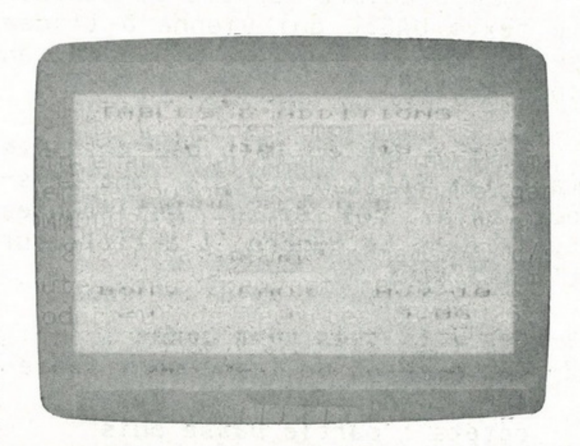
La documentation fournie avec MASTER-SCREEN est bien structurée avec liste des instructions résumées à la fin et renvoyant aux bonnes pages dans la notice. Elle est malheureusement un peu succinte et manque, par conséquent, de détails ou de précisions utiles. Pourquoi par exemple n'est-il rien dit au sujet de l'utilisation de INZ avec une variable numérique ? Dans DECZ, le type M (lettres capitales) pour la zone exige que le VIC soit en mode minuscules/majuscules et non majuscules/graphique, faute de quoi la saisie ne prend que des symboles graphiques......

MASTER-SCREEN est donc une cartouche particulièrement destinée à faciliter la gestion de l'écran de VIC avec organisation de zones ou de fenêtres, de défilements localisés (à une fenêtre), de saisie d'information contrôlée - en particulier les contrôles de types et certains plantages des INPUT -.
Il permet aussi d'aligner des sommes avec francs et centimes, ce qui n'est pas immédiat en BASIC si on ne dispose pas de possibilités analogues à celles de PRINT USING .

La sauvegarde et le chargement de pages d'écran sont particulièrement indiqués aussi pour des applications utilisant des menus.

D. TRECOURT





Deux aspects de l'écran au cours du programme

Comment l'ATARI range les programmes BASIC en memoire

Une des premières choses à savoir lorsqu'on veut analyser le fonctionnement d'un interpréteur BASIC est la façon dont le texte BASIC est stocké dans la mémoire. La façon dont les variables sont codées et rangées en mémoire lui est liée et elle est, elle aussi, importante à connaître. Cette connaissance ouvre des applications intéressantes que nous esquisserons en fin d'article.

Le texte BASIC

La première manière de stocker le texte BASIC qui vienne à l'idée serait de le stocker tel quel, en code ASCII.

Ce n'est en fait pas possible mais les BASIC Microsoft font presque cela. Les seules différences (voir dans ce numéro, l'article sur l'ORIC) sont :

1- les mots clés sont codés
2- les numéros de ligne sont représentés en binaire sur deux octets : partie basse puis partie haute
3- la fin de ligne est marquée par un octet nul
4- les lignes successives sont chaînées entre elles : en tête de chaque ligne figure un lien sur deux octets qui pointe vers la ligne suivante. La fin de programme est marquée par trois octets nuls (fin de ligne + lien nul)

Sinon, le corps de l'instruction, noms de variables, constantes, etc.., est écrit tel quel en ASCII.

Eh bien l'ATARI fait plus de codage que cela. Il code les mots clés bien sûr mais aussi les nombres et les noms des variables.

Le texte BASIC stocké en mémoire est formé de la juxtaposition des codages des différentes lignes. La fin de texte est marquée par la présence implicite d'une ligne de n° 32768.

Structure d'une ligne

Chaque ligne est représentée par une suite d'octets obéissant à la structure suivante :

```
N1, N2, LN, (IS,X,X,...,DL) .....
    (IS,X,X,...,DL)
```

Les deux premiers octets forment le n° de la ligne sous la forme partie basse puis partie haute : donc numéro : N1 + 256 * N2. Le troisième octet LN est la longueur occupée par la ligne, de N1 au dernier DL. Si nous supposons que chaque octet est numéroté de 0 à ..., on voit que le dernier délimiteur aura pour numéro DL-1 et que DL serait en somme le numéro du premier octet de la ligne.

Les parenthèses (le caractère parenthèse ne figure pas dans les octets) sont répétées pour chaque instruction de la ligne (on sait qu'il peut y en avoir plusieurs). A ce moment, le IS d'une instruction est toujours inférieur à LN sauf pour la dernière instruction de la ligne où il vaut LN.

On a représenté par X les octets composant l'instruction. Le dernier, DL, est un délimiteur de fin d'instruction : il vaut 20 sauf celui de la dernière instruction de la ligne qui vaut 27 pour signaler la fin de ligne.

Exemple

Supposons qu'on crée la ligne unique :

260 ? Z : GOTO 12345

on aura en mémoire :

NO octet	valeur	explication
0	4	N1, N2
1	1	260 = 4 + 256 * 1
2	17	adresse ligne suivante
3	7	adresse instruction suivante
4	40	code de ?
5	128	numéro de variable + 128
6	20	fin de la première instruction
7	17	adresse prochaine instruction
8	10	code de GOTO
9	14	annonce un nombre
10	66	exposant
11	1	1 de 1 23 45 (en DCB)
12	35	23 en DCB
13	69	45 en DCB
14	0	chiffres suivants
15	0	s'il y en a
16	22	fin d'instruction et de ligne
17	0	début ligne suivante
18	128	0 + 256 x 128 = 32768 = fin de programme

Cet exemple nous montre comment sont codés les nombres et les instructions et il pose le problème du codage des variables.

Codage des instructions

Chaque mot-clé est représenté par un seul octet, son code, d'où une grande économie de place. Le tableau suivant donne quelques codes. Le programme qui suit vous permet d'afficher le tableau complet des 54 mots-clés avec leur code.

Quelques codes

Code	Mot-clé :
0	REM
7	IF
10	GOTO
11	GO TO
12	GOSUB
13	RESTORE
14	annonce un nombre
18	entre nombres
20	fin d'instruction
22	fin de ligne
23	GOTO dans ON ...GOTO
24	GOSUB dans ON...GOSUB
27	THEN
30	ON
35	TRAP
40	?

Programme pour avoir tous les codes

```
10  DIM A$ (10)
20  A = 42161 : ? "ESC CLEAR"
30  FOR K = 0 TO 53
40  A$ = ""
50  FOR J = 1 TO 10
60  C = PEEK (A) : IF C  128
      THEN 80
70  A$ (J) = CHR$ (C) : A = A + 1
      : NEXT J
80  A$ (J) = CHR$ (C - 128)
90  PRINT K, A$ : A = A + 3
      NEXT K
```

Le programme est basé sur le fait qu'en ROM à partir de l'adresse 42161 il y a la liste des mots-clés en ASCII avec 128 rajouté au code du dernier caractère pour signaler la fin du mot. Le mot-clé est suivi de l'adresse de la routine de traitement de ce mot-clé; on la passe par le A = A + 3 en 90.

Le codage des nombres est fait sous la forme suivante :

octet 14 pour annoncer que les 6 octets suivants forment un nombre, exposant qui fixe la place de la virgule et 5 octets contenant les chiffres du nombre exprimés en décimal codé binaire. On voit qu'il y a place pour 10 chiffres significatifs.

Representation des variables

Dans le corps du programme, les variables sont codées : chaque variable est représentée par un code = 128 qui est en fait 128 + numéro de la variable. Le numéro n'est rien d'autre que le classement des arrivées de chaque variable dans le programme. Ainsi notre variable unique Z porte le n° 0 et elle a pour code 128. Le code devant être = 255 on voit que cela limite à 128 variables distinctes au maximum.

La représentation des variables fait ensuite appel à deux tables. Une première table, la table des noms, contient les noms de toutes les variables bout à bout dans l'ordre d'arrivée dans le programme, donc dans l'ordre des numéros. Les noms sont mis en code ASCII mais pour reconnaître la fin d'un nom, le dernier code est augmenté de 128. Ainsi, si l'on a les variables AB et AC on aura dans la table : 65, 194, 65, 195. Cette façon de faire n'implique aucune limitation dans la longueur d'un nom, c'est pourquoi en BASIC ATARI un nombre variable peut avoir jusqu'à 128 caractères tous significatifs; il n'y aura qu'un octet occupé dans le texte BASIC.

La deuxième table contient les valeurs, là encore dans l'ordre des numéros. Chaque nombre occupe 8 octets :

- le premier est le type de la variable (0 pour les scalaires réels)
- le second est le numéro de la variable
- les six suivants sont analogues (donc exposant et BCD) à la représentation dans le texte BASIC.

Les pointeurs fondamentaux

Oui, mais à quelles adresses tout ceci se trouve-t-il ? Eh bien, les adresses sont variables pour pouvoir fonctionner avec des tailles mémoire diverses. On a donc une gestion par pointeur.

Le couple 136 (partie basse), 137 (partie haute) pointe vers le début de la première ligne BASIC. Ainsi, dans notre exemple de ligne 260: PEEK(PEEK(136)+256*PEEK(137)) donnerait 4.

Le couple 137,138 pointe, lui, vers la ligne en cours d'exécution.

Le couple 134,135 pointe vers le début de la table des variables (valeurs).

Le début de la table des noms de variables est en 2048 pour un système sans disque et en 7676 pour un système avec disque.

Applications

Les applications de ces notions sont nombreuses et importantes. Elles donnent lieu à des "méta-programmes", c'est-à-dire des programmes qui agissent sur un programme. L'exemple le plus célèbre est la renumérotation : il ne suffit pas de changer les numéros en tête de ligne, il faut aussi changer les références qui suivent GOTO GOSUB, etc.., donc, pour cela, il faut parcourir le programme et reconnaître les codes des mots-clés voulus.

Nous le ferons dans un prochain numéro. Sinon, on peut changer les références à des variables, rechercher les lignes qui font référence à une variable donnée (références croisées) etc.., Ces notions sont donc importantes à connaître car à l'origine de manips spectaculaires.

Rita A....

64 à Brac

Quelques ficelles pour 64

- empêcher "stop" : POKE 808,239
  ou bien POKE 788,52

- empêcher list et save :
  POKE 818,32

Numéro du bloc vidéo de 16K adressé (entre 0 et 3):
? 3 - PEEK (56576) AND 3

adresses relatives au bloc vidéo de

-l'écran:
? (PEEK (53272) AND 240)/ 16 * 1024

-du générateur de caractères:
? (PEEK (53272) AND 14) * 1024

-de la carte haute résolution :
? (PEEK (53272) AND 8) * 8192

Pour les jeux en langage machine, il existe un générateur aléatoire ultra-rapide de nombres entre 0 et 255 :

mettre 255 ($FF) en 54287 ($D40F)
puis 128 ($80) en 54290 ($D412)

la case 54299 ($D41B) génère alors à une cadence très rapide des octets aléatoires.

Etant donné que les adresses du 64 et du VIC sont pratiquement les mêmes, on peut reprendre des astuces utilisant les ROM du VIC en tenant compte de la translation

$C000 - $ E000 --> $A000 - $C000
VIC 64

Pour les adresses entre $E000 et $E33B, ajouter 3 pour obtenir l'adresse correspondante sur le 64.

Hervé LE MARCHAND

Conserver le curseur pendant GET :

POKE 204,0

Supprimer le curseur :

POKE 204,1

Le supprimer pendant INPUT :

POKE 207,1

Désactiver STOP/RESTORE :

POKE 808,225
(235 pour le réactiver)

Une erreur-système curieuse

Amenez le curseur en dernière ligne. Faites comme si vous entriez une ligne de programme plus longue que les 40 caractères, donc tapez un numéro puis des caractères jusqu'à faire une remontée de ligne. A ce moment, faites un DEL.

Et voilà,

plantė !

En fait, selon la couleur des caractères en vigueur, lorsque cela se produit, on peut s'en sortir en faisant SHIFT/RUN. Obéissez au message PRESS PLAY ON TAPE puis faites STOP.

Cela marche si la couleur des caractères est noir, blanc, pourpre, vert, orangé, brun, gris moyen ou vert clair. Avec les autres couleurs, donc le bleu standard, le plantage est irrémédiable.

Daniel-Jean DAVID

Stockage des programmes dans la mémoire de l'ORIC

Des renseignements essentiels pour qui veut comprendre le fonctionnement de son interpreteur BASIC .

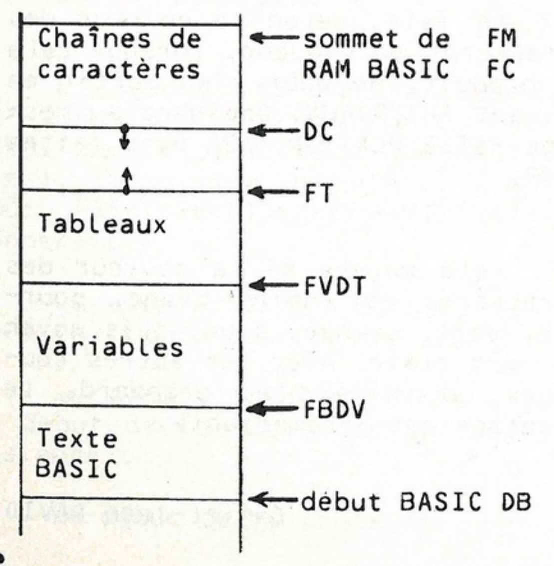
Noter que les informations donnees ici sont exactement valables aussi pour tous les Commodore.

Un certain temps après la sortie du PET (1977), lorsque les fanatiques travaillaient dur pour essayer de percer les secrets de leur machine, la découvere de la façon dont un programme est stocké en mémoire a été le premier élément qui ait soulevé un coin du voile.

Nous espérons que la donnée des mêmes informations sur l'ORIC fera autant de plaisir aux possesseurs d'ORIC. Les principes suivis sont exactement les mêmes que ceux des COMMODORE.

Les pointeurs fondamentaux

En BASIC interprété, on trouve en mémoire à la fois le programme et les variables dans des zones successives, conformément à la figure suivante :



Les adresses, qui sont variables, sont gérées grâce à 7 pointeurs fondamentaux. Leurs adresses pour les COMMODORE ont été données dans divers numéros antérieurs de LA COMMODE. Celles du 64 sont dans ce numéro.

- DB : début de BASIC. Sur ORIC, DB est le couple #9A, 9B (décimal 154, 155) dans l'ordre partie basse, partie haute.Attention, c'est le pointeur.
  La véritable adresse de début de BASIC est DEEK (#9A).
  Usuellement, sur ORIC, on trouve #0501.
- FBDV : fin de BASIC, début des variables. Sur ORIC, c'est le couple #9C, 9D (156, 157)
- FVDT : fin des variables, début des tableaux, #9E, 9F (158, 159)
- FT : fin des tableaux, #A0, A1 (160, 161) L'adresse de fin des tableaux augmente à mesure que le programme s'amplifie.
- DC : début des chaînes, # A2, A3 (162, 163). L'adresse de début des chaînes diminue à mesure que le programme s'amplifie et qu'il apparaît de nouvelles chaînes.
- FC : fin des chaînes, #A4, A5 (164, 165). Ce pointeur est habituellement égal au sui-

vant.

- FM : fin de mémoire vive accessible à BASIC, #A6, #A7 (166, 167). C'est ce pointeur qu'on altère lorsqu'on veut se ménager une zone tranquille en fin de mémoire par l'instruction HIMEM.

Notez que pour bien fonctionner BASIC exige que l'octet juste avant le début de BASIC soit nul. Si vous voulez reculer le début de BASIC à 601 par exemple, il faut faire :

```
POKE 155,6 : POKE 157,6 :
  POKE 159,6
```

mais aussi POKE 6*256,Ø .

Organisation du texte BASIC

A partir de l'adresse DEEK (154), le texte BASIC est formé de la juxtaposition des différentes lignes. Chaque ligne est stockée presque comme elle a été entrée, en ASCII. Les seules différences sont :

1- chaque mot-clé est remplacé par un seul octet, son code. Par exemple, END est codé 128 (#80) GOTO, 151 (#97); IF, 153 (#99); RESTORE, 154 (#9A), etc...,

2- le numéro de ligne est codé en deux octets dans l'ordre partie basse partie haute. Ces deux octets forment les octets 2 et 3 de la ligne (en fait les 3ème et 4ème, car on numérote à partir de 0).

3- l'ensemble des lignes est construit en liste chaînée. Les deux premiers octets (n°s 0 et 1) forment ce qu'on appelle le lien. Ce n'est rien d'autre que l'adresse absolue dans la mémoire du premier octet de la ligne suivante. Le lien est, bien sûr, codé sous la forme partie basse,partie haute.

4- le dernier octet est 00 pour marquer la fin de ligne.

5- la fin du programme est marquée par une ligne fictive à lien nul. Cela fait donc trois octets nuls consécutifs avec la fin de la dernière ligne.

On obéit donc au schéma suivant :

```
     154,155
00   L1 L2 N1 N2 texte 00
     lien
     L'1 L'2 N'1 N'2 texte 00
         :
         :
         :
     L1 L2 n1 n2      .....  00
     00 00
```

Exemple :

Si l'on introduit la ligne unique suivante :

```
260 ? Z : GOTO 12345
```

on aura en mémoire:

adresse	valeur	explication
#501	16	L1 = 10 hex
502	5	L2 :pointe vers 510
503	4	N1 :4+256*1
504	1	N2 : = 260
505	186	code de ?
506	90	"Z"
507	58	":"
508	151	code de GOTO
509	32	"espace"
50A	49	"1"
50B	50	"2"
50C	51	"3"
50D	52	"4"
50E	53	"5"
50F	0	fin de ligne
510	0	nouvelle ligne (vide)
511	0	

Applications

Ces notions ont des applications très intéressantes. On peut (ce n'est pas recommandé) faire qu'un programme se modifie lui-même, par exemple en changeant les noms des variables. On peut faire un programme de renumérotation ou un programme qui examine le texte BASIC et en extrait les références à certaines variables, etc...,

Honoric de BALSA

Les ficelles de La Commode

Pour tous les Commodore

Pas tout-a-fait un Autorun

Vous savez que "SHIFT" "RUN" fait charger le premier programme sur la cassette (ou disque, si vous êtes en BASIC 4.0). N'y a-t-il pas moyen de faire mieux ?

Si ! Vous pouvez obtenir le même effet pour n'importe quel programme, avec les cassettes uniquement.

Vous tapez 4 espaces, guillemet, le nom du programme voulu, guillemet et puis vous revenez en début de ligne par des "curseur gauche" et là, vous faites votre "SHIFT" "RUN".

Qu'en est-il des disques ? Avec BASIC 3.0, et aussi sur VIC et 64, rien ne vous empêche de faire :

4 espaces guillemet d:nom guillemet virgule huit (ou 9..), curseurs à gauche, "SHIFT"-"RUN".

Avec BASIC 4.0, la solution est la suivante :

4 espaces trois "INSERT" trois curseurs gauche (qui se feront en mode guillemet) d : nom curseurs à gauche pour venir en tête de ligne et le "SHIFT" "RUN".

N.B : Ceci n'est pas un Autorun. Dans l'Autorun, le programme se met à s'exécuter dès chargement. Ceci est inscrit sur le disque et la bande et constitue une protection contre les piratages.

Pierre-Etienne THALBERG

Line Feed or not Line Feed

Un des problèmes les plus souvent rencontrés avec la connexion d'imprimantes non COMMODORE (et même certaines COMMODORE, n'est-ce-pas 8027 ?..) est celui du line-feed (alimentation de ligne) automatique: les imprimantes COMMODORE accompagnent tout retour-chariot d'une alimentation de ligne et, bien sûr, les logiciels supposent qu'elles le font. Maintenant, si vous avez une imprimante qui ne le fait pas, vous risquez de tout imprimer sur la mêmeligne! C'est, en particulier, le cas des listings.

Ceci se produit, notamment, avec certaines imprimantes connectées par l'interface RS 232 du VIC ou du 64 (ce n'est pas le seul problème du RS 232 ! ...)

La solution ? Ouvrir un fichier de numéro logique supérieur ou égal à 128. Sauf en BASIC 1.0 où il n' a pas de solution simple, ces fichiers ajoutent automatiquement le line-feed.

Ex. : OPEN 128,2,0,CHR$ (6)
ouvre un fichier RS232 à 300bd
ensuite, pour lister :
CMD 128 : LIST

Daniel-Jean DAVID

Rename disk

Le programme suivant permet de renommer une disquette. Il fonctionne sur tous les systèmes COMMODORE (8000, 4000, 3000, PET, 64 et VIC) avec toutes les unités de disques sauf la 8050.

```
5 PRINTCHR$(147)
6 PRINTSPC(12)"# RENAME DISK #":PRINT
10 OPEN1,8,15,"I0
40 FORI=144TO163
50 PRINT#1,"M-R"CHR$(I)CHR$(7)
60 GET#1,A$
70 PRINTA$;
80 NEXTI
100 PRINT:PRINT:INPUT"NOUVEAU NOM ";N$
110 N$=LEFT$(N$+"",18)
120 PRINT:INPUT"NOUVEL INDENTIFICATEUR ";ID$
130 CH$=N$+LEFT$(ID$,2)
140 PRINT#1,"M-W";CHR$(144)CHR$(7)CHR$(20)CH$
150 PRINT#1,"B-A:0,18,0
160 PRINT#1,"I
999 CLOSE1
```

Cyril CAMBIEN

Les Disquettes La Commode

Ca-y-est! La Commode commercialise des disquettes. D'abord, vous pouvez avoir toute cassette La Commode sous forme de disquette (supplément 30F).

Pour CBM (64 en préparation):

VIDEO de P. CASTRATARO
Un systèmee de gestion de votre cassettothèque (jusqu'à 1000 cassettes). Recherche multicritères notamment en fonction du temps restant. Prix: 310F

REPERTOIRE ET AGENDA de P. CASTRATARO
Gestion de votre agenda et de votre carnet d'adresses. Recherche complète ou par initiales - mot de passe. Sortie écran ou imprimante. Prix: 310F

REL.FIC2 de P. CASTRATARO
Gestion de base de données fichiers relatifs. Prix: 310F

PCM8000 de P. CASTRATARO
Transforme votre 8000 en terminal performant qui se connecte à tous réseaux (Transpac, Missive, etc...) à l'aide d'un modem accoustique. Transfert facile de fichiers. Prix: 2500F

Pour VIC et 64

BD64 de D.J. DAVID
Gestion d'une base de données sur VIC ou 64 équipé de sa 1541. Prix: 290F

BON DE COMMANDE

nom: . . . .
adresse: . . . .
. . . .

Je désire les disquettes (entourer) format 4040/1540 8050

VIDEO REPERTOIRE REL.FIC2

PCM8000 BD64 signature:

(règlement joint obligatoirement)

Elargir l'affichage du VIC

L'affichage de 22 colonnes par ligne est ressenti par la plupart comme une des plus graves limitations du VIC. De nombreux lecteurs nous ont écrit pour nous demander des renseignements sur des extensions permettant de faire passer l'affichage du VIC à 40 et même 80 colonnes par ligne. Faisons le point.

1 - Il est vrai que de telles extensions existent effectivement et permettent de passer même à 80 colonnes. Il faut noter d'ailleurs que si le 40 colonnes est utilisable avec un téléviseur, le mode 80 colonnes exige, lui, un moniteur de bonne qualité: sinon ces extensions se placent facilement sur le BUS (connecteur des cartouches d'extension).

2 - Aucune de ces extensions n'est et ne sera en vente en France. Il faut se les procurer en Angleterre.

3 - Compte tenu du fonctionnement du VIC - on l'a vu dans les tentatives d'élargir légèrement l'affichage (LA COMMODE, n° 5, p. 32) - ces extensions ne peuvent marcher qu'en :

- se substituant complètement au circuit d'affichage du VIC,
- transformant les routines de gestion de l'affichage.

Ceci implique un prix élevé. De fait, on dépasse même le prix du VIC.

4 - Il en résulte que la solution "vente du VIC - achat d'un C64" est certainement bien plus efficace : moins coûteuse, moins hasardeuse et en plus on profite des avantages du 64.

Pierre-Etienne THALBERG

Montage sonore pour le 64

C'est dans notre prochain numéro que nous aborderons la programmation du synthétiseur sonore du 64: en effet, il reste des points à éclaircir sur son fonctionnement, pour obtenir des résultats reproductibles. Il semble que l'ordre des POKE joue: les essais à faire sont très nombreux et, comme vous le savez, LA COMMODE ne veut donner que des indications parfaitement éprouvées et fiables.

En attendant, le système sonore du 64 se heurte à un problème très gênant : il est affecté d'un important bruit de fond, de nature à masquer complètement ses qualités sonores.

Un premier remède, lorsque vous utilisez une prise Péritel consiste à incorporer un filtre très simple dans le circuit son:

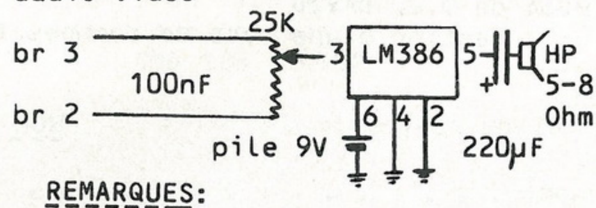
<u>64</u> <u>Péritel</u>

100K
3 —— 6
47nF
2 —— 17

Le condensateur et la résistance trouvent place dans la prise Péritel elle-même.

Un remède plus radical consiste à ne pas utiliser le son de la télé : le ronflement sera fortement éliminé. Nous proposons un montage complet d'ampli sonore et haut-parleur. Il est utile notamment à ceux qui utilisent un moniteur dépourvu de son. Il fait appel à un circuit intégré amplificateur LM 386 : on le trouve chez tous les revendeurs, notamment chez TANDY.

connecteur
audio video



<u>REMARQUES</u>:

1 . Les broches 1, 7 et 8 du LM 386 sont non connectées dans ce montage. Un condensateur entre 1 et 8 augmente le gain.
2 . C'est précisément un LM 386 qui est utilisé dans l'ORIC pour piloter le haut parleur interne.

Daniel-Jean DAVID

JOUEZ AVEC VOTRE COMMODORE

60 NOUVEAUTES VIC 20 et 64

Catalogue gratuit sur demande





TICSOFT =

Des jeux, des outils d'aide à la programmation, des accessoires, des périphériques, bibliothèque française, étrangère.

TICSOFT sélectionne chaque mois les meilleures nouveautés et vous propose sur cette page un échantillonnage de son catalogue gratuit.

VIC 20

• VICBUS 3 + - Carte d'extension pour 4 cartouches dont 3 sélectionnables. Fusible et bouton de reset ..... **490,00 F** TTC

KAKTUS - Des guêpes, des frelons attaquent un cactus géant .... **190 F** TTC

C 64

SIREN CITY - Véritable course poursuite opposant policiers et gangsters dans une cité dangereuse............. **146 F** TTC

VIZAWRITE - Traitement de texte (AZACC[E]) avec rom et étiquettes pour clavier. **Prix incroyable** .... **1.355** TTC

BUS - C 64 - Carte d'extension pour 4 cartouches **sélectionnables.** Fusible et reset (poussoir) .... **682** TTC





TICSOFT

est un département de

ic informatique center

Résidence Jacquard
5, rue Ovigneur - 59800 LILLE
BP 1029 - Tél. (20) 54.61.01

DEMANDEZ NOS PRODUITS CHEZ VOTRE REVENDEUR, OU UTILISEZ LE BON DE COMMANDE CI-DESSOUS.

☐ Je désire recevoir le catalogue gratuit TICSOFT 10/83
Je possède : ☐ VIC 20
☐ C 64

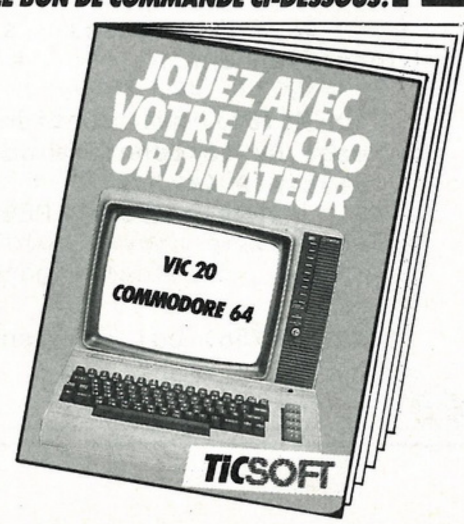
M. ____________________

N° __________ Rue ____________________

CP ⌊⌊⌊⌊⌊⌊ Ville ____________________

Qt	Titre	Prix unit.	TOTAL TTC
		+ Forfait PORT et EMBALLAGE	15,00
		TOTAL TTC DE MA COMMANDE	

Signature



que je règle par chèque joint à l'ordre de TICSOFT / Informatique Center.
* Toute commande non accompagnée du règlement ne pourra être honorée.

1 CASSETTE GRATUITE "TICSOFT" POUR TOUT ACHAT DE 4 PRODUITS

Inventez votre propre jeu de caractères sur ATARI

La possibilité pour l'utilisateur de définir son propre jeu de caractères ouvre des applications très intéressantes :

- définition d'éléments de dessins pour certains graphiques

- définition d'objets, insectes ou envahisseurs pour des jeux

- définition de caractères spéciaux alphabet grec, signes mathématiques, caractères accentués pour un traitement de textes.

Cette possibilité est exclue pour les CBM : le générateur de caractères est en ROM et il n'est pas déplaçable. Pour l'ORIC, le générateur de caractères est d'office en RAM (après une copie, à partir de la ROM, évidemment). Il est donc à volonté modifiable.

Sur les VIC et 64 le générateur est normalement en ROM mais son origine est modifiable et on peut la définir de façon que le générateur soit en RAM. Eh bien l'ATARI a exactement la même possibilité. Pour les VIC, 64 et ORIC, ceci a été expliqué dans les livres "LA DECOUVERTE ..." correspondants.

Pour l'ATARI, nous le décrivons ici. Dans la suite, on appellera G l'origine du générateur de caractères. Lorsque celui-ci est en ROM, G = 57344 ($E000).

Deux adresses stratégiques interviennent :

. HMEM(106) qui contient le n° de page de la dernière adresse ROM possible (+1) :
    DERAD= 256 * PEEK (106)

et

. CBASE(756) qui contient le n° de page du début du générateur de caractères : G = 256 * PEEK(756)

Les étapes à suivre sont :

1 - Choisir l'emplacement RAM où on mettra le générateur de caractères. Le problème est le suivant : entre G et DERAD on doit trouver le générateur de caractères et la mémoire d'écran (cf.fig.1). La taille du générateur est 1K (128 caractères x 8 octets par caractère).
Celle de l'écran varie avec le mode d'affichage : dans les trois modes texte elle est

avec	GRAPHICS 0	992 octets
avec	GRAPHICS 1	674
avec	GRAPHICS 2	424

Comme G doit être à une frontière de K (comme DERAD), on voit que le mieux est :
G = DERAD - 2048

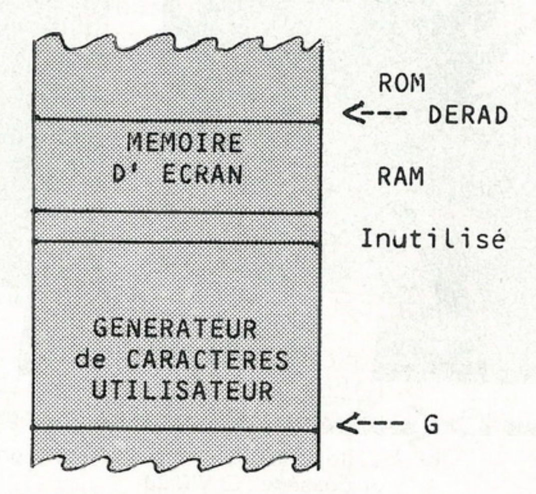


fig1 Carte mémoire

2 - Copier le générateur de caractères en ROM dans celui en RAM, car il y a bien des caractères normaux que vous voudrez conserver. Cette étape peut être supprimée si vous

ne voulez garder aucun caractère standard, ou elle peut être abrégée si vous en gardez peu.

On peut utiliser la séquence :

```
10 GROM = 256 * PEEK (756)
20 G = 256 * PEEK (106) - 2048
30 FOR I = 0 TO 1023
40 POKE G + 1, PEEK (GROM + 1)
50 NEXT I
```

3 - Définir les nouveaux caractères. Chaque caractère se définit dans une matrice 8x8 dont chaque ligne correspond à un octet dans le générateur de caractères. Chaque point "allumé" forme un bit égal à 1, voir fig. 2

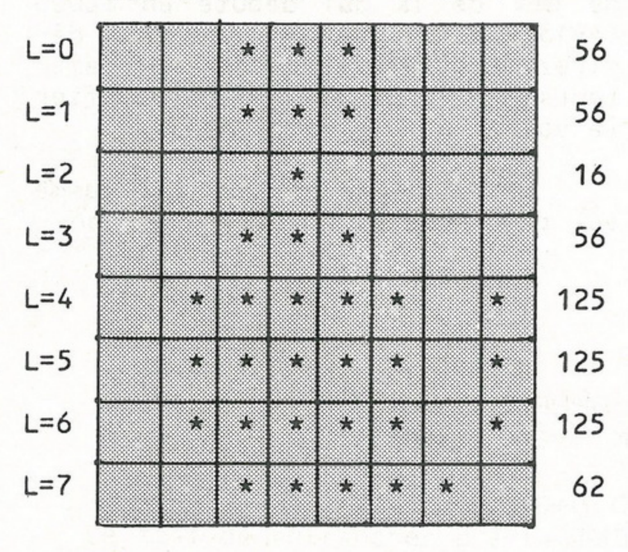


fig.2 Définition d'un caractère

Si R est le code écran du cacaractère à définir (R va de 0 à 127), l'octet correspondant à la ligne élémentaire n° L (L va de 0 à 7) est à l'adresse : G + 8 * R + L. La valeur se définit en sommant, sur une grille analogue à la fig.2, les nombres 1,2,4 ...128 correspondant aux points allumés. On a marqué sur la fig.2 les valeurs correspondant au petit chat dessiné. Par exemple, le 56 de la ligne L = 0 représente 8 + 16 + 32.

Les données des caractères peuvent être groupées dans des DATA. Mais il y a mieux.

Le programme suivant constitue un mini éditeur de caractères:

```
60 DIM R$ (2), K$ (2), A$ (8)
70 INPUT "CARACTERE A REMPLACER";R$
     : R = ASC (R$) - 32
80 PRINT "ESC CLEAR";
90 FOR L = 0 TO 7 : PRINT
     "  ........" : NEXT L
100 PRINT "ESC curseur haut
     (8 fois)" ;
110 FOR L = 0 TO 7
120 INPUT A$ : A = 0
130 FOR K = 1 TO 8 : K$=A$(K,K)
   : IF K$ = "." THEN 150
 140 A = A + 2^ (8 - K)
150 NEXT K : A = INT (A + 0.4)
160 POKE G + 8 * R + L,A : NEXT L
170 PRINT "ESC CLEAR" : PRINT
   "      " ; R$
```

l'aide de ce programme, vous définissez votre caractère en grand, ligne par ligne : pour les points éteints,vous passez les points sur l'écran par mouvements de curseur. Pour les points allumés, vous remplacez le point par une étoile par exemple.

La dernière instruction est sensée imprimer le caractère obtenu. En fait, elle imprime le caractère standard dont votre caractère prend la place. Vous n'aurez votre caractère qu'une fois accomplie la dernière étape.

4 - Dire à l'ATARI qu'on utilise le nouveau jeu de caractères. Ceci est accompli par :

```
180 POKE 756, G / 256
```

Et voilà ! En rassemblant les différents morceaux du programme donnés ici (la section 70 - 170 peut être répétée) vous vous dotez d'un système efficace vous permettant de gérer très facilement les dessins les plus divers grâce à 128 caractères programmables.

Rita A...

Fonction APPEND pour 64 (et VIC)

Le programme ci-dessous, écrit pour un CBM 64, permet de placer en mémoire centrale deux programmes l'un au bout de l'autre. Il s'agit donc de l'équivalent de la fonction APPEND à laquelle on a recours, en particulier, lorsqu'un programme est conçu selon une suite de sous-programmes stockés séparément sur cassette ou disque. Le moment crucial consiste à réunir tous les sous-programmes pour ne faire plus qu'un programme principal. Pour cela, la fonction APPEND est naturellement la bienvenue.

Le programme comporte 34 octets de langage machine qui peuvent être placés dans une zone tranquille de la mémoire. Ici, le tampon du magnétophone à cassette n'est pas très propice au cas où vous utilisez ce périphérique. Aussi, la zone de MEV de 1K qui débute en $C000 (49152) a été choisie. Si vous désirez stocker le programme ailleurs, il vous suffit de modifier la valeur de N à la ligne 30.

Le principe de ce programme est très simple; en effet, on con-

```
100 REM **** FONCTION APPEND ****
110 REM ****  POUR CBM - 64   ****
120 :
130 N=49152:FOR L=N TO N+33
140 READ A:POKE L,A:NEXT
150 :
160 DATA 56,162,0,134,191,165,45,233
170 DATA 2,176,4,162,255,134,191,133
180 DATA 43,24,165,46,101,191,133,44
190 DATA 96,169,1,133,43,169,8,133,44
200 DATA 96
210 :
220 A$=CHR$(34)
230 PRINT "▮▮    FONCTION APPEND
240 PRINT "      ******** ******
250 PRINT "▮1 - LOAD "A$"PROGRAMME 1"A$
260 PRINT "▮2 - SYS"N
270 PRINT "▮3 - LOAD "A$"PROGRAMME 2"A$
280 PRINT "▮4 - SYS"N+25
290 PRINT "▮     C'EST TOUT !
```

sidère que le début de la mémoire BASIC se trouve à la fin du premier programme en mémoire, auquel on ajoute le second par l'instruction LOAD. Il suffit maintenant de redonner au pointeur de la mémoire BASIC sa valeur d'origine soit $0801 (2048) et le tour est joué !

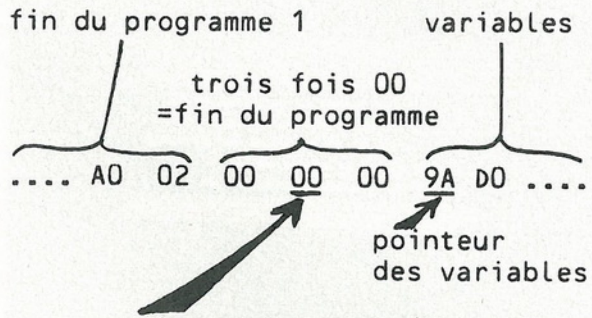
Les pointeurs modifiés sont :

Début de MEV BASIC	43 ($2B)	
	44 ($2C)	
Fin du programme, début des variables	45 ($2D)	
	46 ($2E)	

Il est possible de simuler la fonction APPEND au moyen des ordres directs ci-dessous :

```
LOAD "PROGRAMME 1"
A=PEEK(45)-2:B=0
IF A<0 THEN A=256+A:B= 1
POKE 43,A:POKE 44,PEEK(46)-B
LOAD "PROGRAMME 2"
POKE 43,[1]:POKE 44,(8)
```

Le fait de retrancher 2 à la valeur du pointeur de variables s'explique de la manière suivante :



L'inconvénient du mode direct est naturellement le nombre de manoeuvres à effectuer. Avec le programme machine, tout devient plus simple :

```
FONCTION APPEND
******** ******
1 - LOAD "PROGRAMME 1"
2 - SYS 49152
3 - LOAD "PROGRAMME 2"
4 - SYS 49177
    C'EST TOUT !
```

Si vous possédez un VIC 20, tout n'est pas perdu, puisque deux petites modifications suffisent pour que le programme tourne sur votre machine.

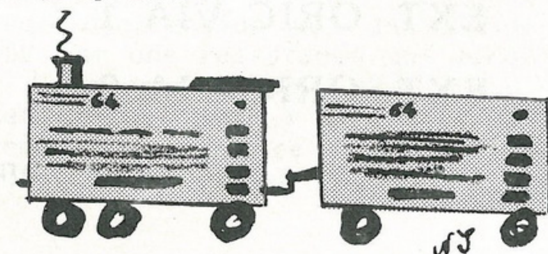
Pour cela, procédez comme suit:

PRINT PEEK (43) puis notez la valeur que vous placez dans le carré (ligne 190)

PRINT PEEK (44) ici, le résultat doit remplacer la valeur contenue dans le cercle.

Ces corrections permettent de donner au programme la valeur du début de la mémoire BASIC qui varie selon la machine. Si vous utilisez le programme machine, n'oubliez pas de modifier la valeur de N (ligne 130). Au cas où aucun endroit de la mémoire ne se prête à garder sans problème le programme, il est toujours possible de donner au pointeur de fin de mémoire (en 55 et 56) une valeur légèrement inférieure, ce qui libère quelques octets pour stocker le programme machine.

Enfin, pour ceux qui désirent plus de détails, voici la liste de ce programme



```
.,  C000 38        SEC
.,  C001 A2 00     LDX #$00
.,  C003 86 BF     STX $BF
.,  C005 A5 2D     LDA $2D
.,  C007 E9 02     SBC #$02
.,  C009 B0 04     BCS $C00F
.,  C00B A2 FF     LDX #$FF
.,  C00D 86 BF     STX $BF
.,  C00F 85 2B     STA $2B
.,  C011 18        CLC
.,  C012 A5 2E     LDA $2E
.,  C014 65 BF     ADC $BF
.,  C016 85 2C     STA $2C
.,  C018 60        RTS
.,  C019 A9 01     LDA #$01
.,  C01B 85 2B     STA $2B
.,  C01D A9 08     LDA #$08
.,  C01F 85 2C     STA $2C
.,  C021 60        RTS
```

49152

Après le LOAD "PROGRAMME 1", on effectue la correction du pointeur en 43-44 (début de BASIC) qui prend la valeur de celui en 45-46 (début des variables) moins 2.

49177

Le pointeur en 43-44 retrouve sa valeur d'origine après le LOAD "PROGRAMME 2".

Voilà, j'espère que ce programme pourra rendre service à de nombreux propriétaires de CBM 64 ou de VIC 20 toujours avides de trucs qui augmentent les possibilités de leur machine.

Pierre MENETRE

Architecture de l'ORIC

(Premiers éléments)

Il y a beaucoup moins à dire sur l'architecture de l'ORIC que sur celle du 64 : les problèmes de superposition de mémoire sont nettement plus simples. C'est plutôt au niveau des entrées/sorties que réside une certaine complexité que cet article s'efforce de clarifier.

Les boitiers

L'ORIC est construit autour d'un nombre extrèmement réduit de boîtiers essentiels.

Il y a d'abord deux processeurs qui se partagent le bus (1):

Le microprocesseur qui est un 6502 est le coeur du système; c'est lui qui exécute les instructions. Le second processeur appelé ULA (Universal Logic Array) est un processeur d'entrées-sorties spécialisé, conçu en propre par ORIC. C'est lui qui gère l'écran, mais en plus, et c'est la source de la simplicité de construction et donc du bas prix de l'ORIC, il prend en charge toutes les fonctions auxiliaires, notamment le décodage d'adresses, diminuant de beaucoup le nombre de boîtiers annexes nécessaires.

Cette structure à deux processeurs est maintenant universelle dans les micros bon marché. Elle a été introduite par Sinclair sur le ZX 81 puis utilisée par COMMODORE sur le VIC et le 64.

----

(1) Pour une définition de ces mots tels que "bus", "processeurs", "boîtiers d'E/S)", "sélections de boîtier", etc.., se reporter par exemple à "SYSTEMES A MICROPROCESSEURS"de D.J. DAVID, chez EDITESTS.

Au point de vue mémoire ROM, tout réside dans un boîtier unique 23128 de 16K octets (c'est le summum de la technique actuelle) aux adresses C0)0-FFFF. Dans les versions préliminaires, il était remplacé par deux EPROM 2764 de 8K octets, d'où la présence de deux emplacements sur la carte.

La mémoire RAM est formée de 8 boîtiers de RAM dynamique 4164 (64K x 1) pour le 48K,ou 2 boîtiers 4416 (16K x 4) pour le 16K.

Il y a enfin deux boîtiers d'entrées-sorties : un VIA 6522 qui fournit 2 ports de 8 bits parallèles et deux temporisateurs et un GI AY-3-8912 qui est le synthétiseur sonore. Seul le VIA est sur le bus. Le synthé. est accédé par un port du VIA.

Règlement du conflit mémoire

C'est l'ULA qui fournit les sélections de boîtiers aux mémoires et aux entrées-sorties. Comme l'espace adressable du 6502 est limité à 64K, dans la version 48K, on ne peut pas profiter totalement des 64K de RAM présents sur la carte. Parmi ces 64K, 16K sont occultés au profit de la ROM.

Pour cela, lorsque l'adresse est entre 0 et $BFFF, l'ULA sélectionne la RAM. Lorsqu'elle est entre C000 et FFFF, l'ULA sélectionne la ROM et inhibe la RAM. Il y a une broche appelée $\overline{MAP}$ sur le connecteur d'extension qui est tirée à +5V par une résistance qui aboutit à la broche 26 de l'ULA. Si l'on met cette broche à 0, il est vraisemblable qu'à ce moment, l'ULA inhibe la ROM et sélectionne la

RAM. On disposerait alors de 64K de RAM. Nous parlons au conditionnel car ORIC n'a fourni aucune information sur le fonctionnement de l'ULA.

Il y a une autre broche du même genre sur le connecteur d'extension, appelée ROMDIS, qui aboutit à une sélection de boîtier active haute de la ROM. Si vous la mettez à 0, vous inhibez la ROM interne et vous pouvez alors activer une ROM externe, contenant par ex. un autre langage que BASIC.

Les entrées-sorties

Seul le VIA est sur le bus aux adresses $300-30F. C'est un port du VIA qui sert à accéder au synthétiseur 8912, ce qui n'est pas grave car le synthétiseur fournit lui-même un port de 8 bits parallèles. Voici les assignations :

Port A du VIA (adresse: $301,
  registre direction: $303) :
- port parallèle pour l'interface imprimante
- accès au synthé (avec,en plus,CA2 et CB2 : dans un prochain numéro,nous décrirons comment accéder directement au synthé)
- CA1 fournit l'entrée ACK de l'imprimante

Port B du VIA (adresse: $300,
  registre direction: $302)
- PB0 à PB2 : rangée clavier
- PB3 : sélection clavier
- PB4 : signal STROBE de l'imprimante
- PB5 : inutilisé
- PB6 : télécommande magnéto
- PB7 : sortie magnéto
- CB1 : entrée magnéto

Port du synthétiseur :
  colonnes clavier .

Une broche analogue à ROMDIS appelée I/O-Control apparaît sur le connecteur d'extension : en la mettant à 0 elle permet d'inhiber le VIA et d'activer les entrées-sorties propres à l'utilisateur. Sachant qu'il y a décodage incomplet des adresses d'E/S sur le signal I/O, on peut d'ailleurs conserver le VIA interne et en ajouter à l'extérieur.

Divers

Le bouton appelé RESET l'est improprement. En fait, il crée une interruption NMI. En revanche, la broche appelée RESET sur le connecteur d'extension est bien liée au signal RESET.

Daniel-Jean DAVID

Vérification de cassette sur ATARI

Introduction

On ne sait jamais si une cassette s'est trouvée bien écrite ou non. Les COMMODORE possèdent une instruction (VERIFY) qui permet de vérifier si une sauvegarde s'est bien effectuée alors que le programme à sauver est encore en mémoire. En cas d'incident, on peut refaire la sauvegarde.

On peut faire presque la même chose sur l'ATARI avec le petit programme suivant, qui relit la cassette et signale les incidents de lecture. On pourrait d'ailleurs pousser plus loin et faire - comme les COMMODORE - une comparaison entre les valeurs trouvées sur la bande et les valeurs en mémoire.

Le programme lit chaque bloc de la cassette et, par appel à une routine en langage machine, transfère ce bloc dans une chaîne de 128 caractères A$, toujours la même, implantée à l'adresse pointée par $203, 204.

Programme BASIC

Il faut, bien sûr, que le premier numéro de ligne de votre programme soit supérieur à 10. Vous changez en 0 le 255 de la ligne 4 si vous voulez sauver votre programme par SAVE au lieu de CSAVE.

En 9 se trouve le DATA qui définit la routine en langage machine de transfert du tampon cassette vers la variable A$. Voici cette routine :

Partie en langage machine

Elle est implantée en $600, ce qui est habituel sur l'ATARI.

Ce programme met en évidence deux adresses stratégiques du système de l'ATARI; le tampon cassette : $400 et suivantes et le nombre d'octets contenus dans le tampon : $028A. (le listing est sur la page suivante).

Rita A...

```
1 CLR:DIM A$(128):GOSUB 8
2 FOR I=1536 TO 1564: READ A :POKE I,A :NEXT I :TRAP 6
3 PRINT"ESC CLEAR";"METTEZ LA CASSETTE ET PRESSEZ UNE TOUCHE"
4 CLOSE #1: OPEN #1,4,255,"C:"
5 GET #1,A:X=USR(1536): GOSUB 8: GOTO 5
6 IF PEEK(195)=136 THEN CLOSE #1:PRINT "CORRECT": END
7 PRINT "ERREUR NO";PEEK(195);" MAUVAISE CASSETTE": END
8 POKE 203,ADR(A$)-(INT(ADR(A$)/256)*256): POKE 204,INT(ADR(A$)/256): RETURN
9 DATA 104,174,138,2,134,61,160,0,162,0,185,0,4,129,203,200,230,203,208,2,230,
      204,196,61,240,2,208,238,96
```

```
                *  = 1536
0600 68            PLA
0601 AE 8A 02      LDX $028A        ;TAILLE DU
                                    ;TAMPON CASSETTE
0604 86 3D         STX $3D
0606 A0 00         LDY #$00
0608 A2 00         LDX #$00
060A B9 00 04      LDA $0400,Y      ;PREND UN
                                    ;OCTET DU TAMPON
060D 81 CB         STA ($CB,X)      ;MET EN A$
060F C8            INY              ;SUIVANT
0610 E6 CB         INC $CB          ;INCR BAS
0612 D0 02         BNE $0616
0614 E6 CC         INC $CC          ;INCR HAUT
0616 C4 3D         CPY $3D          ;TOUT TRANS
0618 F0 02         BEQ $061C        ;OUI:RETOUR
061A D0 EE         BNE $060A        ;NON:BOUCLE
061C 60            RTS
```

Complétez votre Collection

Les numéros anciens de La Commode sont encore disponibles mais peut-être pas pour longtemps. Hâtez vous de compléter votre collection si vous ne voulez pas rater des renseignements essentiels.

BON DE COMMANDE

à envoyer à La Commode 28, rue Vicq-d'Azir 75010 PARIS

Je désire les numéros suivants de La Commode (entourer)

1 2 6 7 8

Joignez votre réglement (45F par numéro) à l'ordre de SEDERMI

☐ CCP ☐ CB ☐ MANDAT

Bientôt épuisés

NOM ....

ADRESSE ....

....

....

Programmathèque

Solitaire

AUTEUR : A. HESBOIS

SYSTEME : CBM 3032 + EDEX 2.0

ADAPTATIONS : Modifier L et U ligne 18:

PET 2001	CBM 4000	CBM 8000
L = 245		
U = 58843	U = 57471	U = 57447

REMARQUE: Ne résout pas le solitaire, mais vérifie les coups du joueur et déplace les pions selon ses ordres. En début de partie, il met tous les pions en place, vous épargnant cette tâche fastidieuse: quelle aubaine !

```
11 REMI--------------------I
12 REMI   LE SOLITAIRE     I
13 REMI  (C) A. HESBOIS    I
14 REMI   LE  13/03/83     I
15 REMI--------------------I
16 :
17 REM INITIALISATIONS
18 L=216:U=57949:X=33176:DIMA(8,8)
19 FORI=1TO20:A$=A$+"  ":B$=B$+"■■":NEXT
20 FORI=0TO1:FORJ=0TO1:A(I,J)=2:A(I,J+5)=2:A(I+5,J)=2:A(I+5,J+5)=2
21 FORI=0TO6:FORJ=0TO6:IFA(I,J)<>2THENA(I,J)=1                :NEXTJ,I
22 NEXTJ,I:A(3,3)=0
23 :
24 REM PRESENTATION
25 PRINT"■"B$;:PRINT"■LE SOLITAIRE"SPC(13)"(C) A. HESBOIS■":PRINTB$;
26 PRINT"■TOUS LES PIONS DOIVENT DISPARAITRE .":PRINT"■SAUF UN !!!
27 PRINT"■POUR PRENDRE UN PION,IL FAUT PASSER PAR ■DESSUS AVEC UN AUTRE .
28 PRINT"■LES PRISES SE FONT VERTICALEMENT OU      ■HORIZONTALEMENT .■
29 PRINTB$;:PRINT"■BONNE CHANCE !!! C'EST DUR,DUR .■":PRINTB$:GOSUB69
30 :
31 REM AFFICHAGE JEU
32 PRINT"■LES COUPS SONT ENTRES COMME LES EXEMPLESSUIVANTS :
33 PRINT"PION LIGNE 2 COLONNE 3 = 23":PRINT"PION LIGNE 0 COLONNE 4 = 04
34 PRINT"■POUR ARRETER APUYER SUR ■ RUN/STOP "
35 POKEL,8:SYSU:PRINTTAB(8)"0 1 2 3 4 5 6
36 PRINTTAB(6)"■0 . . ● ● ● . . 0
37 PRINTTAB(6)"1 . . ● ● ● . . 1
38 PRINTTAB(6)"2 ● ● ● ● ● ● ● 2
39 PRINTTAB(6)"3 ● ● ●   ● ● ● 3
40 PRINTTAB(6)"4 ● ● ● ● ● ● ● 4
41 PRINTTAB(6)"5 . . ● ● ● . . 5
42 PRINTTAB(6)"6 . . ● ● ● . . 6":PRINTTAB(8)"■0 1 2 3 4 5 6
43 :
44 REM JEU
45 GOSUB65:FORI=20TO22:POKEL,I:SYSU:PRINTA$;:NEXT
46 POKEL,20:SYSU:INPUT"PION A DEPLACER....■■■00■■■■■";PD:CA=PD:GOSUB73
47 IFER=1THEN46
48 LD=LS:CD=CS:IFA(LD,CD)<>1THEN52
49 POKEL,21:SYSU:INPUT"CASE OU VA LE PION.■■■00■■■■■";CP:CA=CP:GOSUB73
50 IFER=1THEN49
51 LC=LS:CC=CS:IFA(LC,CC)=0THEN53
52 BEEP15,15:POKEL,22:SYSU:PRINT"COUP IMPOSSIBLE !":GOTO45
53 IFLC=LDAND(CC=CD-2ORCC=CD+2)THEN56
54 IFCD=CCAND(LC=LD+2ORLD=LC+2)THEN56
55 GOTO52
56 LO=LD:CO=CC:IFLD>LCTHENLO=LD-1
57 IFLD<LCTHENLO=LC-1
58 IFCD>CCTHENCO=CD-1
59 IFCD<CCTHENCO=CC-1
60 IFA(LO,CO)=0THEN52
61 A(LC,CC)=1:A(LD,CD)=0:POKEX+CC*2+LC*40,81:POKEX+CD*2+LD*40,32
62 A(LO,CO)=0:POKEX+CO*2+LO*40,32:GOTO45
```

```
63 :
64 REM TEMPO1
65 TE=TI
66 IFTI-TE<15 THEN 66:ELSERETURN
67 :
68 REM TEMPO2
69 TE=TI
70 IFTI-TE<300THEN70:ELSERETURN
71 :
72 REM VERIFICATION CASE
73 ER=0:IFCA>64ORCA<2ORCA<>INT(CA)THENBEEP15,15:ER=1:RETURN
74 LS=INT(CA/10):CS=CA-10*LS:RETURN
```

Finish

AUTEUR : C.CAMBIEN

SYSTEME : CBM 64

ADAPTATIONS : aucune, trop spécifique du CBM 64

REMARQUE : le mode d'emploi est donné par le programme

```
1 REM FINISH 1.2 - CBM 64 - PAR C.CAMBIEN - 06/1983
2 REM ################################################
3 GOSUB5000
4 REM INITIALISATION DU JEU
5 POKE53281,12
10 DIMX(7),Y(7)
20 PRINTCHR$(147)CHR$(142)
25 VL=8.5:VI=0:AC=3:IN=1
30 VOL=54296:AD=54277:W=54276:HI=54273:LO=54272:SR=54278
50 FORI=1TO23:PRINTTAB(31)"I":NEXT
60 PRINT"■■"TAB(32)"SCORE:■■■
70 PRINTTAB(32)"VITESSE:■■
80 PRINTTAB(32)"TEMPS:■■■■■■■
90 PRINTTAB(32)"■■FINISH■■■
92 PRINTTAB(32)"C.C.1983
100 V=53248:POKEV+21,0
120 FORI=0TO62:READQ:POKE832+I,Q:NEXT
125 FORI=0TO62:READQ:POKE896+I,Q:NEXT
130 FORI=2040TO2045:POKEI,13:NEXT:POKE2046,14
150 FORI=0TO5:READX(I):READY(I):NEXT
200 POKEV+23,127:POKEV+29,127:POKEV+28,63:POKEV+37,11:POKEV+38,1:POKEV+39,1
205 FORI=37TO44:READQ:POKEV+I,Q:NEXT
210 FORI=0TO6:POKEV+2*I,X(I):POKEV+2*I+1,Y(I):NEXT
220 POKEV+16,0:POKEV+21,63:POKEVOL,5:SC=0:T=TI
500 REM PROGRAMME PRINCIPAL
505 A=PEEK(203):IFA<>64THENGOSUB1000
510 GOSUB2000
520 A=PEEK(203):IFA<>64THENGOSUB1000
570 GOSUB3000
580 SC=SC+VI+12
582 IFSC<0THENSC=0
585 PRINT"■■■■"TAB(32)SC"■  ":PRINT"■■■"TAB(32)INT(VI*5)+100"■ "
587 PRINT"■■■"TAB(32)INT((TI-T)/60)
589 POKEW,0:POKEAD,97:POKESR,128:POKEW,33:POKEHI,VI/3+7:POKELO,VI/3+8
590 IFTI-T<3600THEN500
600 REM FIN DE LA PARTIE - AFFICHAGE DU          SCORE ET DU RECORD
605 POKEVOL,0:POKEV+21,0:PRINT"■VOTRE TEMPS EST TERMINE
610 S$="S.":IFSC=0THENS$=".
630 PRINT:PRINT"VOTRE SCORE EST DE"SC"POINT"S$
640 IFSC>BSTHENBS=SC
```

```
641 S$="S."'IFBS=0THENS$=".
642 PRINT:PRINT"LE MEILLEUR SCORE EST DE"BS"POINT"S$
650 PRINT:PRINT"▓APPUYER SUR SHIFT POUR REJOUER.
660 WAIT653,1:RESTORE:GOTO20
1000 REM DEPLACEMENT ET AFFICHAGE DE LA          VOITURE DU JOUEUR
1010 IFA=36THENIFX(0)<221THENX(0)=X(0)+VL:GOTO1500
1020 IFA=39THENIFX(0)>24THENX(0)=X(0)-VL:GOTO1500
1030 IFA=10THENIFVI<30THENVI=VI+AC:GOTO1500
1040 IFA=12THENIFVI>-10THENVI=VI-AC
1500 POKEV+30,0:POKEV,X(0)
1510 IFPEEK(V+30)<>0GOTO4000
1520 RETURN
2000 REM DEPLACEMENT DES VOITURES          ADVERSES
2010 FORI=1TO5
2020 Y(I)=Y(I)+RND(TI)*(VI+2)-IN
2040 IFY(I)>240THENY(I)=0:NEXT:RETURN
2050 IFY(I)<0THENY(I)=255
2070 NEXT:RETURN
3000 REM AFFICHAGE DES VOITURES ADVERSES
3005 POKEV+30,0:FORI=1TO5
3010 POKEV+2*I+1,Y(I):NEXT
3020 IF(PEEK(V+30)AND1)<>1THENRETURN
4000 REM ACCIDENT => EXPLOSION
4005 POKEVOL,15:POKEW,0:POKEAD,28:POKESR,0:POKEW,129:POKEHI,17:POKELO,37
4009 POKEV+12,X(0):POKEV+13,Y(0)
4010 POKEV+21,127:FORK=1TO5:FORL=0TO15:POKEV+45,L:NEXTL,K
4020 POKEV+21,63:SC=SC-3*(VI+100)
4030 POKEVOL,5:RETURN
5000 REM INSTRUCTIONS
5005 POKE53281,8
5010 PRINTCHR$(147)CHR$(14)CHR$(5)
5020 PRINT:PRINTTAB(12)"## ─╮╭╮♥| 1.2 ##
5030 PRINT:PRINT"  ×OUS ETES LE PILOTE D'UNE VOITURE DE
5040 PRINT"DE COURSE ET VOUS DEVEZ PARCOURIR LA
5050 PRINT"PLUS GRANDE DISTANCE POSSIBLE EN UNE
5060 PRINT"MINUTE.
5070 PRINT:PRINT"  \AIS IL Y A D'AUTRES VOITURES SUR LA
5080 PRINT"PISTE ET VOUS DEVEZ EVITER LE PLUS
5090 PRINT"POSSIBLE LES COLLISIONS.
5100 PRINT:PRINT"  ×OICI VOS COMMANDES :
5110 PRINT:PRINT"- ♠ POUR ACCELERER.
5120 PRINT"- ♦ POUR RALENTIR.
5130 PRINT"- / POUR ALLER A GAUCHE.
5140 PRINT"- \ POUR ALLER A DROITE.
5150 PRINT:PRINT:PRINT"       ▓♣PPUYEZ SUR SHIFT POUR JOUER.
5160 WAIT653,1:RETURN
10000 REM CODAGE DU SPRITE 'VOITURE'
10005 DATA0,12,0,0,12,0,1,63,16,1,191,144,1,55
10010 DATA16,0,55,0,0,55,0,0,63,0,0,63,0,0,255,192,0,251,192
10040 DATA0,234,192,0,230,192,0,213,192
10050 DATA20,213,197,20,247,197,22,221,229
10060 DATA20,127,69,20,64,69,0,64,64
10070 DATA0,0,0
10500 REM CODAGE DU SPRITE 'EXPLOSION'              152,6,93,32,1
10505 DATA0,0,0,0,8,0,0,136,128,4,73,16,2,73,32,1,42,64,8,170,
10510 DATA190,192,0,127,0,31,255,252,0,127,0,1,190,192,6,93,32
10520 DATA64,2,73,32,4,73,16,0,13,6,128,0,8,0,0,0     ,8,170,152,1,42
11000 REM POSITIONS INITIALES DES          VOITURES
11010 DATA153,200,19,0,70,100,121,25,172,0,223,75
12000 REM COULEUR DE LA CABINE DES          VOITURES
12010 DATA11,1,7,3,4,5,6,8
```

Symetries

AUTEUR : Jacques VAISSIERE

SYSTEME : Les CBM 40 colonnes

ADAPTATION :

Pour le 64, l'adaptation est assez facile, mais l'absence de clavier numérique rend les mouvements moins intuitifs. Il serait préférable d'utiliser un manche à balai.

Sinon,

ligne 100	A = 1024	
120	CC = 1433	
290	CC = 1433	
300	CC = 1455	
310	CC = 2015	
490	(CC - 1064)	(2 fois)
500	(CC - 1048)	(2 fois)

Pour CBM 8000 ou VIC, il faudrait repenser l'affichage en fonction de la largeur de ligne différente, notamment modifier les 39, 40, 41 des lignes 380 à 500.

REMARQUES :

CC : Adresse écran du dernier point allumé

300-320	Impression de * au départ
370-520	Déplacement de la balle
530	DATA pour le dessin programmé
540-680	Présentation du clavier numérique
690	Détermine le n° de ligne et de colonne suivant la valeur de CC
700	Détermine l'adresse écran du point à imprimer pour la symétrie ortho.
710	Même chose mais pour la symétrie point
730	Teste la validité du déplacement
780-830	Si le déplacement n'est pas valide clignotement

Ce programme permet de présenter la symétrie orthogonale et la symétrie centrale à un niveau élémentaire.

En mode démonstration, les déplacements que vous choisissez font dessiner la balle et son symétrique.

```
 100 A=32768
 110 GOSUB540:PRINT"♡"
 120 CV=81: CC=33177: PRINT"IL S'AGIT D'
ETUDIER LES SYMETRIES"
 130 PRINT:PRINT"-1- SYMETRIE ORTHOGONAL
E"
 140 PRINT:PRINT"-2- SYMETRIE CENTRALE"
 150 PRINT:INPUT"QUE CHOISIS-TU";XX
 160 PRINT:PRINT: INPUT"OPTION DEMONSTRA
TION (O-N)";F$:PRINT:PRINT
 170 IFF$="O"THENPRINT"♡":CV=87:     GOTO2
00
 180 PRINT"OPTIONS:-1- DESSIN PROGAMME
                 -2-         ALEATOIRE"
 190 PRINT:PRINT:INPUT"QUE CHOISIS-TU";G
G :PRINT"♡"
 200 ONXXGOSUB 840,860:IFF$="O"THEN 290
 210 V=17:IFGG=2THENV=25
 220 POKECC,42:N=87
 230 FORH=1TOV
 240 X=INT(9*RND(.25))+1:IFGG=1THENREAD
X
 250 Y=INT(5*RND(.25))+3:IFGG=1THENREAD
Y
 260 FORJ=1TOY:GOSUB370:NEXT
 270 NEXT
 280 IFXX=2THENPOKEA+500,170
```

```
 290 CC=33177:POKECC,42
 300 IFXX=1THENCC=33199
 310 IFXX=2THENCC=33359
 320 POKECC,42:N=81
 330 GETA$:IFA$=""THEN330
 340 IFA$="#"THENRESTORE:PRINT"♥":F$="":
GOTO120
 350 X=VAL(A$):GOSUB370:GOSUB 690
 360 GOTO330
 370 ON X GOTO380,390,400,410,440,420,43
0,450,460
 380 Z=39:CC=CC+Z:GOTO470
 390 Z=40:CC=CC+Z:GOTO470
 400 Z=41:CC=CC+Z:GOTO470
 410 Z=-1:CC=CC+Z:GOTO470
 420 Z=1 :CC=CC+Z:GOTO470
 430 Z=-41:CC=CC+Z :GOTO470
 440 GOTO470
 450 Z=-40:CC=CC+Z :GOTO470
 460 Z=-39:CC=CC+Z :GOTO470
 470 IFINT((CC-A)/40) =(CC-A)/40THENCC=C
C-Z:GOTO520
 480 IFCC<AORCC>A+999THENCC=CC-Z:GOTO520
 490 IFINT((CC-32808)/40) =(CC-32808)/40
THENCC=CC-Z:GOTO520
 500 IFN=87ANDINT((CC-32792)/40)=(CC-327
92)/40THENCC=CC-Z:GOTO520
 510 POKECC,N
 520 RETURN
 530 DATA 8,3,9,3,3,6,2,4,1,5,2,4,6,4,8,
4,7,2,4,10,7,4,6,3,8,1,7,3,8,8,6,4,2,2
 540 PRINT"♥QQQQ"
 550 PRINTTAB(14)" 7 8 9 "
 560 PRINTTAB(15)"  "
 570 PRINTTAB(14)"4 5 6"
 580 PRINTTAB(15)"  "
 590 PRINTTAB(14)" 1 2 3 "
 600 PRINT"VOUS VOUS TROUVEZ AU CENTRE D
U
 610 PRINTTAB(10)"CLAVIER NUMERIQUE
 620 PRINT:PRINT"    POUR VOUS DEPLACER,T
APEZ LE CHIFFRE
 630 PRINT"Q CORRESPONDANT A LA DIRECTI
ON CHOISIE
 640 PRINT:PRINT"     VOUS PILOTEZ●"
 650 PRINT:PRINT"TAPEZ # POUR RECOMMENCE
R":PRINT
 660 PRINT"QQ TAPEZ SUR UNE TOUCHE POUR
COMMENCER
 670 GETA$:IFA$=""THEN670
 680 RETURN
 690 CO=INT((CC-A)/40):LI=CC-A-CO*40
 700 IFXX=1THEN DD=A+(CO-1)*40+80-LI
 710 IFXX=2THENDD=A+(23-CO)*40+80-LI
 720 IFF$="O"THEN760
 730 IFPEEK(DD)<>87ANDPEEK(DD)<>81 THENG
OSUB780
 740 IFPEEK(DD)=87THENF=0
 750 IFF=1THEN770
```

```
760 POKEDD,CV
770 RETURN
780 FORI=1TO15:POKECC,32
790 FORJ=1TO50:NEXT:POKECC,209
800 FORK=1TO50:NEXT
810 NEXT
820 POKECC,32:CC=CC-Z:F=1  :L=L+1
830 RETURN
840 FORI=1TO24:POKEA+20+40*I,93:NEXT
850 RETURN
860 POKEA+500,170:RETURN
```

Marienbad

AUTEUR : D. FLEURY

SYSTEME : VIC et tous systèmes CBM, 64.., puisqu'il n'utilise aucune particularité du VIC.

REMARQUE : L'utilisateur joue contre le programme. Le programme demande au joueur sa force, s'il veut commencer, et s'il veut une configuration de départ non standard.

```
10 DIMK(9),KI(9):T$="JEU DE MARI
ENBAD"
20 GOSUB1000
30 GOSUB1400
40 GOSUB1600
50 GOSUB1300:GOSUB1100
60 PRINT"■■VOULEZ-VOUS COMMENCER
"
70 INPUT"OUI OU NON";R$:IF R$="O
UI" THEN240
80 IF R$<>"NON" THEN70
90 REM PROG. JOUE
100 X=0:I1=-1
110 FORI=0TON:X=X AND NOT K(I) O
R NOT X AND K(I)
120 IF K(I)>P-L AND P>L OR L=1 T
HEN I1=I:Y=L-INT(L/2)*2
130 NEXT
140 IF I1>=0 THEN 220
150 I1=0:IFX=0THEN 200
160 Y=X AND NOT K(I1) OR NOT X A
ND K(I1):IF Y<=K(I1) THEN 180
170 I1=I1+1:GOTO160
180 G=(L-1)*(P-2)*(10-F)/40-1
190 IF RND(1)*4>G THEN 220
200 I1=INT(RND(1)*(N+1)):IF K(I1
)=0 THEN 200
210 Y=INT((K(I1))*RND(1)↑(2/F))
220 GOSUB 1200
230 IFL=1 AND P=1 THEN355
240 REM JOUEUR...
250 PRINT"■■A VOUS DE JOUER■"
260 INPUT"NO. DE LIGNE";I:I1=I-1

270 IF I1<0 OR I1>N THEN 290
280 IF K(I1)>0 THEN 300
290 PRINT"IMPOSSIBLE":GOTO260
300 PRINT"COMBIEN OTEZ-VOUS DE
PIONS:"
310 PRINT"DE 1 A";K(I1);:INPUTX:
X=INT(X)
320 IF X<1 OR X>K(I1) THEN 310
330 Y=K(I1)-X:GOSUB1200
335 IFP=0THEN355
340 IF L=1 AND P=1 THEN PRINT"■■
BRAVO VOUS AVEZ GAGNE":NG=NG+1:G
OTO 360
350 GOTO100
355 PRINT"■■DESOLE VOUS AVEZ PER
DU":NP=NP+1:GOTO360
360 PRINT"VOUS AVEZ FAIT:"
370 PRINTNG;TAB(5);"PARTIES GAGN
ANTES"
380 PRINTNP;TAB(5);"PARTIES PERD
ANTES"
390 PRINT"■UNE AUTRE PARTIE?"
400 INPUT"OUI OU NON";R$
410 IF R$="NON" THEN PRINT"ALORS
 SALUT!":END
420 IF R$<>"OUI" THEN 400
430 PRINT"■VOULEZ-VOUS CHANGER:■
"
440 PRINT"■I■:LA DISPOSITION INI
TIALE SEULE?"
450 PRINT"■F■:LA FORCE SEULE?"
460 PRINT"■T■:LES DEUX?"
470 PRINT"■R■:RIEN DU TOUT?■"
480 INPUT"I,F,T OU R";R$
490 IF R$="I" THEN GOSUB 1400:GO
TO 50
500 IF R$="F" THEN 40
510 IFR$="T" THEN 30
520 IF R$="R" THEN 50
530 GOSUB 1000:GOSUB 1300
540 PRINT"DISPOSITION PRECEDENTE
■":GOSUB 1110:PRINT
550 GOTO430
1000 PRINT"■"T$:RETURN
```

```
1090 PRINT"■"T$:GOTO1105
1100 GOSUB 1000
1105 PRINT
1110 FOR I=0 TO N
1120 PRINTI+1;TAB(5);
1140 FOR J=1 TO 10
1150 IF J>K(I) THEN PRINT" .";:GO
TO1170
1160 PRINT".";
1170 NEXT:PRINT:NEXT:PRINT:PRINT
:RETURN
1199 REM AFFICHE CHANGEMENT CLIG
NOTANT
1200 GOSUB 1100:Z=K(I1)
1210 FORT=0 TO 5:K(I1)=Z:GOSUB10
90
1220 K(I1)=Y:GOSUB 1090:NEXT
1230 P=P+Y-Z:L=L+(Y=0):RETURN
1300 REM:COPIE KI DANS K
1310 P=0:FOR I=0 TO N:K(I)=KI(I)
:P=P+K(I):NEXT
1320 L=N+1:RETURN
1400 PRINT"■4 LIGNES DE 1,3,5 ET
 7PIONS CONVIENNENT ?"
1410 INPUT "OUI OU NON";R$
1420 IF R$="NON" THEN 1460
1430 IFR$<>"OUI" THEN 1410
1440 N=3
1450 FOR I=0 TO N.KI(I)=2*I+1:NE
XT:RETURN
1460 PRINT"DONNEZ LE NOMBRE DE"
PRINT"LIGNES:";
1470 INPUT"DE 2 A 10";L
1480 IFL>10 OR L<2 THEN 1470
1490 N=L-1
1500 PRINT"POUR CHAQUE LIGNE DON
-NEZ LE NOMBRE DE PIONSDESIRES D
E 1 A 10:"
1510 FOR I=0 TO N
1520 PRINT"LIGNE NO";I+1;:INPUT
KI(I)
1530 IF KI(I)<1 OR KI(I)>10 THEN
 INPUT"DE 1 A 10";KI(I):GOTO1530

1540 KI(I)=INT(KI(I)):NEXT
1550 RETURN
1600 PRINT"■FORCE DU PROGRAMME?"

1610 INPUT"DE 1 A 10";F.IF F<1 O
R F>10 THEN1610
1615 T$=LEFT$(T$,16)+STR$(F)
1620 RETURN
```

Variable

AUTEUR : O. BUTLER

SYSTEME : VIC et tous les CBM

REMARQUE : liste les noms des variables qui ont été utilisées effectivement pendant l'exécution du programme que l'on veut tester.

. Le programme "VARIABLE" doit être chargé en même temps que le programme à tester. Il vaut mieux que les numéros de lignes des 2 "programmes" soient différents...

MISE EN OEUVRE :

Après avoir rentré le programme à tester, il faut entrer VARIABLE soit directement au clavier, il n'y a alors aucun problème d'adaptation, soit à partir d'une sauvegarde sur cassette ou sur disque. Dans ce dernier cas, il faut:

1. Modifier l'adresse de début de BASIC pour mettre "VARIABLE" à la suite du programme à tester:

tous VIC et 64:
```
POKE 43,PEEK(45)-2
POKE 44,PEEK(46)
```

CBM:
```
POKE 40,PEEK(42)-2
POKE 41,PEEK(43)
```

2. Charger normalement à partir de la cassette ou du disque la copie de "VARIABLE".

3. Remettre les bonnes valeurs dans le pointeur de début de BASIC:

```
tous VIC et 64: POKE 43,1
+
VIC de base : POKE 44,16
VIC + 3K    : POKE 44,4
VIC + >=8K  : POKE 44,18
 64  : POKE 44,8

CBM : POKE 40,1
      POKE 41,4
```

4. Exécuter le programme à tester.

5. GOTO 10001

Et la liste des noms des variables effectivement utilisées s'affiche, avec en prime celles de "VARIABLES".

```
10001 V=PEEK(45)+256*PEEK(46)
10002 V1=PEEK(47)+256*PEEK(48)
10003 FORI=VTOV1+35
10004 K=PEEK(I):K1=PEEK(I+1)
10005 IFK<128ANDK1<128THENM$="":GOTO10008
10006 IFK>=128ANDK1>=128THENM$="%":K=K-128:K1=K1-128:GOTO10008
10007 IFK<128ANDK1>=128THENM$="$":K1=K1-128:GOTO10008
10008 P$=CHR$(K):N$=CHR$(K1)
10009 PRINTP$N$M$
10010 I=I+6
10011 NEXT
```

Duplic

AUTEUR : La Commode

SYSTEME : Tout système Commodore avec unité à double platine 3040, 4040 ou 8050. Marche aussi pour un 64 équipé de l'interface IEEE parallèle.

MISE EN OEUVRE :
Répondre aux questions:
"type d'unité" (3040, 4040 ou 8050)
"n° de périph" (8 le plus souvent ou 9, 10, ...)
"platine départ,platine arrivée"
ex. 0,1 signifie qu'on copiera le disque monté en 0 sur le disque monté en 1. Se tromper à cet endroit est catastrophique.

```
10 PRINT"■■■ DUPLICATION BRUTALE DE DISQUES"
20 PRINT"              (C) LA COMMODE"
30 PRINT"LES DISQUETTES DOIVENT ETRE FORMATEES"
40 PRINT"■N.B.■ CERTAINES COPIES SONT ILLEGALES
50 PRINT"  LA COMMODE N'EST PAS RESPONSABLE"
60 PRINT"DES INFRACTIONS AUX DROITS DES TIERS"
70 PRINT"■■"
80 INPUT"TYPE D'UNITE ";T
90 INPUT"NO. DE PERIPH. ";U
100 PRINT"PLATINE DE DEPART,PLATINE D'ARRIVEE ?"
110 PRINT"■(NE VOUS TROMPEZ PAS)■";
120 INPUT D,A
130 IF T=3040 GOTO 1100
140 GOSUB 500:IF T=4040 GOTO1100
150 GOSUB 500:GOTO 1100
500 FOR I=1 TO 12:READ X:NEXT:RETURN
1100 OPEN80,U,15,"I":OPEN82,U,2,"#"
1200 FOR Z=1 TO 4
1300 READ P1,P2,S2
1400 FOR P=P1 TO P2
1500 FOR S=0 TO S2
1600 PRINT#80,"B-R"2;D;P;S
1700 PRINT#80,"U2:"2;A;P;S
1900 NEXT:NEXT:NEXT:CLOSE80
2000 END
3040 DATA 1,17,20,18,24,19,25,30,17,31,35,16
4040 DATA 1,17,20,18,24,18,25,30,17,31,35,16
8050 DATA 1,39,28,40,53,26,54,64,24,65,77,22
```

ORIC à BRAC

Chargement d'une zone memoire

Au chapitre consacré à cette possibilité, intéressante au demeurant, le manuel d'utilisation omet de signaler une particularité assez gênante.

En effet, au chargement à partir d'une cassette le pointeur de fin de BASIC prend la valeur de la fin de la zone, telle qu'elle a été définie lors de l'enregistrement.

Inutile de dire que le programme BASIC déjà en mémoire ne pourra alors plus tourner normalement : généralement, on obtient le message d'erreur : OUT OF MEMORY après l'ordre RUN.

Donc, si on doit charger une zone mémoire : langage machine, écran, jeu de caractères, etc.., le faire en premier. Charger ensuite le programme BASIC. (Il n'est pas nécessaire de faire NEW).

En cas d'impossibilité :
Sauvegarder le pointeur de fin du programme BASIC par PRINT DEEK(#9C) et le rétablir après chargement de la zone mémoire par

DOKE #9C, xxxx

N.B. Les zones mémoire peuvent aussi être sauvegardées en vitesse lente comme les programmes :

CSAVE "ZONE", Axxxx, Eyyyy, S

Cela n'est pas dans le manuel.

Niveau magneto

Il semble qu'une partie des ennuis qu'on a avec les cassettes soit due à ce que l'ORIC a un niveau de sortie trop fort, qui convient plutôt à une entrée AUX mais pas à une entrée MICRO.

On peut corriger très simplement en insérant une résistance de 50 à 100 K en série dans la sortie : la mettre du côté ORIC, le mieux est dans la prise DIN,

Jean RENAUD

Saisie au clavier

Trois adresses sont concernées #208, #2DF, et #35.

#208 contient en permanence l'état du clavier (56 si aucune touche n'est enfoncée, sinon un code qui représente l'emplacement géographique et non le code ASCII de la touche enfoncée)

#2DF contient le code ASCII de la dernière touche enfoncée + 128. Elle contient 0 au départ.

#35 contient le code ASCII de la dernière touche enfoncée, mais seulement après un GET. Ainsi :

```
GET A$ : ? ASC (A$) et
GET A$ : ? PEEK (#35)
```

sont équivalentes.

Seule #208 évolue en temps réel et donc permet la saisie au vol. #2DF ne change que lors de

l'appui d'une nouvelle touche, ou alors elle est remise à 0 lors d'un GET. #35 n'est chargée que s'il y a un GET.

Honoric de BALSA

Une anomalie de la fonction STR$

Tout se passe comme si le STR$ d'un nombre positif commençait par un caractère nul non imprimé. Celui-ci compte dans LEN(STR$(X)) et dans toutes les fonctions de manipulation de chaîne. En fait, le caractère ajouté est &STR$ (2) de telle sorte que si vous faites PLOT X,Y, STR$(N), le nombre est imprimé en vert !!!

Problemes avec certaines imprimantes

Comme tous les systèmes analogues, l'ORIC fonctionne par interruptions régulières tous les soixantièmes de seconde. Ces interruptions peuvent troubler le fonctionnement de certaines imprimantes, notamment la SEIKOSHA. En effet, celle-ci a besoin de recevoir rapidement les caractères à imprimer. Or, l'ORIC est incapable de les fournir quand il est occupé par une interruption. La solution ? Inhiber les interruptions avant LPRINT et les rétablir après. Cela se fait par :

```
CALL #ED01 : LPRINT .....
     CALL #ECC7
```

Pour un listing, seul le premier CALL est à faire car LLIST fait stopper l'exécution de la ligne. On reprend le contrôle en faisant RESET.

Conversions Hexadecimal-decimal

La conversion de décimal à hexadécimal ne pose aucun problème, que le nombre à convertir soit une constante ou une variable; on fait:

```
? HEX$ (10000) ou  ? HEX$ (A)
```

Dans l'autre sens, pour une constante, il suffit de faire p.ex. ? #FFFF . Mais si le nombre hexa à convertir est une variable A$? Eh bien, on fait :

```
     ? VAL ("#" + A$)
```

BASIC, dear WATSON !

Daniel-Jean DAVID

Limite superieure de la zone BASIC

Selon le manuel, le sommet de la RAM pour les programmes est fixé à l'adresse #9800 (sauf si appel est fait à l'instruction GRAB).

Or au démarrage le pointeur de fin de BASIC est initialisé à #9F00. Cela est sans inconvénient en mode TEXT. Par contre, si on passe en mode HIRES. Il y a écrasement des chaînes de caractères qui sont venues se loger en haut de RAM. En effet les deux jeux de caractères sont initialisés en HIRES de #9800 à #A000.

Si donc un programme comporte une partie en mode HIRES, il faut impérativement mettre au début l'instruction HIMEM #97FF.

Il semble même que l'initialisation du pointeur de fin de BASIC n'est effective que si il y a eu entrée d'un programme. Essayez juste après la mise en route d'entrer en mode direct des chaînes de caractères (A$ = "abc..." par ex.). Elles vont être stockées en descendant depuis #BFFF et vous les verrez apparaître progressivement en bas de l'écran dès #BFED atteint !

Encore l'apostrophe

Dans le premier ORIC à BRAC, on a évoqué l'anomalie rencontrée avec l'apostrophe qui n'est pas correctement pris en compte dans une entrée avec GET.

Voici deux remèdes. On peut avec, après la ligne 10 GET A$, utliser:

20 IF PEEK (#35) = 39 THEN A$ = "'" (ou A$ = CHR$(39))

ou encore

20 IF LEN (A$) <> 1 THEN A$ = "'"

Curieusement en effet toutes les touches délivrent sur un GET une chaîne de longueur unité (y compris les touches de fonctions) sauf précisément l'apostrophe.

Jean RENAUD

Les sons en mode direct

Dans "La Découverte de l'Oric", D.J. DAVID signale que les canaux 2 et 3 ne sont pas accessibles en mode direct, seul le canal 1 l'est. C'est vrai. Il suffit de faire un PLAY 7,0,0,0 pour ensuite pouvoir utiliser tous les canaux.

Honoric de BALSA

Ecriture de donnees sur cassette

Une des plus graves parmi les rares lacunes du BASIC de l'ORIC est l'absence d'instructions permettant de sauver des données sur cassette au même titre que les programmes. Le programme suivant est le squelette d'une gestion de fichiers sur cassette. Lorsqu'on fait RUN (le magnéto étant en écriture) on acquiert au clavier des chaînes de caractères qu'on transfére sur cassette. On termine en fournissant $$$ comme chaîne. Lorsqu'on fait RUN 40, les données sont relues et imprimées sur l'écran.

En 1000-1010, on écrit la chaîne de caractères A$ en #400 et suivantes,

En 1020-1030, on initialise les adresses stratégiques de gestion des cassettes,

#5F -#60 : adresse de début de la zone mémoire à écrire,

#61 -#62 : adresse de fin,

#63 0 = non autoRUN
 1 = autoRUN

#64 0 = BASIC
 1 = données ou langage machine

#67 0 = rapide
 1 = lent .

Le nom du fichier est en #35 et suivantes, terminé par un octet nul. En 1040, on fabrique le nom "A".

En 1050, écriture du fichier: chaque appel à 1000 crée un "fichier".
En 1060, lecture du fichier.

Pour l'utilisation de ce programme, il est préférable de disposer de la télécommande du magnéto.

Pierre-Etienne THALBERG

```
10 INPUT A$
20 GOSUB 1000:GOSUB 1050
30 IF A$<>"$$$" GOTO 10
35 END
40 GOSUB 1030:GOSUB 1060
50 A$="":A=1024
60 B=PEEK(A):IF B=0 GOTO 80
70 A$=A$+CHR$(B):A=A+1:GOTO 60
80 PRINT A$:IF A$<>"$$$" GOTO 40
90 END
1000 LL=LEN(A$)+1:FOR I=1 TO LL-1
1010 POKE 1023+I,ASC(MID$(A$,I,1)):NEXT:POKE 1023+LL,0
1020 DOKE #5F,#400:POKE #61,LL:POKE #62,4
1030 DOKE #63,#100:POKE #67,0
1040 DOKE #35,65:RETURN
1050 CALL #E6CA:CALL #E57B:CALL #E804:RETURN
1060 CALL #E6CA:CALL #E4A8:CALL #E804:RETURN
```

Super Yam's pour VIC

Une adaptation demandée par de nombreux lecteurs

Le "listing" du programme SUPER-YAM tel qu'il est donné dans le n° 6 ne peut s'appliquer directement au VIC 20. Les modifications suivantes sont indispensables :

- pour cadrer les 5 dés dans les 22 colonnes du VIC : mettre 1 "curseur droit" au lieu de 3 dans les lignes 300 et 2820.

- pour cadrer les tableaux de marques dans ces 22 colonnes, il faut resserrer les intitulés et la largeur des colonnes de résultats. Pour cela, les lignes 3000, 3005, 3010, 3020, 3040, 3060, 3070, 3080, 3120, 3180, 3190, 6000, 6005, 6010, ont été redéfinies.

- De la même façon, pour cadrer les tableaux dans les 23 lignes du VIC, j'ai supprimé les lignes 3090 et 3140 du programme. Cela ne gêne pas la présentation des résultats.

Vous trouverez ci-après le "listing" de ce nouveau programme utilisable sur le VIC. Les lignes 150 à 198 ont été rajoutées pour expliquer le jeu aux non-initiés.

Pour une meilleure mise en page du texte au cours du jeu, les lignes 10, 20, 280, 2800, 2900 ont été corrigées par suppression de "curseurs droits" ou d'espaces (SPC).

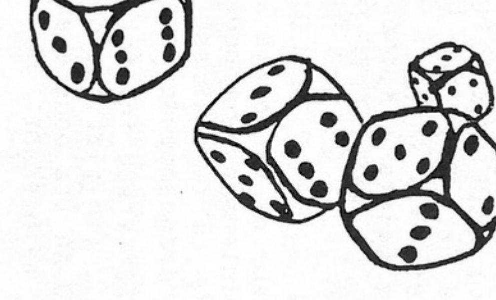
J'ai redéfini la ligne 323 pour rétablir la valeur du paramètre "L" (nombre de dés) à 5, afin de rendre possible un tirage au sort sur 5 dés au 3ème lancer quand le joueur le désire. Cette disposition est rarement possible dans les programmes que vous avez donnés. En effet, au 3ème lancer - initié par la ligne 323 - c'est la valeur de L générée au 2ème lancer qui est réutilisée. Si cette dernière est inférieure à 5, seuls les dés choisis au 2ème lancer seront relancés au 3ème, alors que le joueur a demandé un tirage sur 5 dés.

A. ROUSSEAU

```
10 PRINT"▮▮▮▮▮▮▮▮JEU DE YAM▮"
20 PRINT"▮▮▮▮▮▮==========="
30 INPUT"NOMBRE DE JOUEURS";M
40 FORI=1TOM:PRINT"NOM DU JOUEUR"I;:INPUTM$(I)
50 IFLEN(M$(I))<6THENM$(I)=M$(I)+" ":GOTO50
60 NEXT
140 N=3:A$(1)="MONTEE":A$(2)="DESCENTE":A$(3)="LIBRE"
150 PRINT"▮▮▮CE JEU SE JOUE AVEC 5 DES ET 3 LANCERS.COMBINAISONS
                                                  POSSIBLES:"
160 PRINTSPC(3)"-DE 1 A 6:MAXI DE  DES DU NO,VALEUR:SOMME DES DES
                                                  DU NO CHOISI"
170 PRINTSPC(3)"-FULL:30 PTS"
180 PRINTSPC(3)"-CARRE:30 PTS"
```

```
182 PRINTSPC(3)"-QUINTE:40 PTS"
184 PRINTSPC(3)"-YAM:50 PTS"
186 PRINTSPC(3)"-MAXI:TOTAL DES DES LE PLUS ELEVE POSSIBLE"
187 PRINTSPC(3)"-MINI:IDEM,MAIS INFERIEUR AU MAXI"
188 PRINT"■POUR SUITE TAPER 1 TOUCHE"
190 GETP$:IFP$=""THEN190
192 PRINT"■■AVOIR 61 OU PLUS AU TOTAL DE 1 A 6 DONNE UN BONUS DE 30 PTS"
194 PRINT"■■■CE YAM PEUT SE JOUER  SUR 3 TABLEAUX:"
196 FORI=1TO3:PRINTSPC(5)A$(I):NEXTI
197 PRINT"■■■POUR SUITE TAPER 1 TOUCHE"
198 GETP$:IFP$=""THEN198
200 DIMZ$(12),C%(M,3,12),T%(M,3,12),JJ(36),G(12),X$(6),J(12),C$(3,12)
202 DIMCS(3),CX(3),S$(3),CZ(M,3),CT(3),W$(3),U$(3),LL(M),KK(M)
210 FORJ=1TO12:READZ$(J):NEXT
220 DATA"■■■1■■","■■■2■■","■■■3■■","■■■4■■","■■■5■■","■■■6■■"
222 DATAFULL  ,CARRE ,QUINTE,YAM   ,MAXI  ,MINI  ,
240 X$(1)="■    ■■■■ ● ■■■■    ":X$(2)="■●   ■■■■    ■■■■  ●"
245 X$(3)="■●   ■■■■ ● ■■■■  ●":X$(6)="■●●●■■■■    ■■■■●●●"
250 X$(4)="■● ●■■■■    ■■■■● ●":X$(5)="■● ●■■■■ ● ■■■■● ●"
255 FORA=1TOM:LL(A)=12:KK(A)=1:NEXT
260 FORA=1TOM:FORJ=1TO12:FORI=1TON:C%(M,I,J)=0:T%(M,I,J)=0:NEXT:NEXT:NEXT
261 FORJJ=1TO36
262 FORA=1TOM
270 KM=5:L=5
275 FORH=1TO5:F(H)=H:NEXTH
280 FORK=1TO3:PRINT"■■JOUEUR "M$(A):PRINT:PRINT"COUP NUMERO"JJ:PRINT
285 PRINT"LANCER NUMERO"K"■■■"
287 FORW=1TO15:IFW>1THENPRINT"■■"
288 T%=PEEK(143)
290 FORH=1TOL:B(F(H))=INT(6*RND(T%+H)+1):NEXTH
300 FORD=1TOKM:PRINT"■"X$(B(D))"■■";:NEXTD:PRINT"":NEXTW:PRINT"■■"
305 GOSUB2900
310 IFK=3GOTO350
320 PRINT"■■COMBIEN DE DES REJOUEZ-VOUS? "
321 GETL$:IFL$<>"0"ANDL$<>"1"ANDL$<>"2"ANDL$<>"3"ANDL$<>"4"ANDL$<>"5"
                                                         THEN321
322 PRINTL$:IFL$="0"THEN350
323 IFL$="5"THENL=5:FORH=1TO5:F(H)=H:NEXT:GOTO340
330 L=VAL(L$):PRINT"■QUELS DES ?":FORE=1TOL
331 GETF$
332 PRINTF$SPC(VAL(F$));:IFF$<>"1"ANDF$<>"2"ANDF$<>"3"ANDF$<>"4"ANDF$<>
                                                         "5"THEN331
334 F(E)=VAL(F$)
335 NEXTE:PRINT""
340 NEXTK
350 PRINT"■■SUR QUEL TABLEAU JOUEZ-VOUS(M,D,L)"
351 GETWZ$
352 IFWZ$<>"L"ANDWZ$<>"M"ANDWZ$<>"D"THEN351
355 IFWZ$="L"THEN400
360 IFWZ$="M"THEN364
362 IFWZ$="D"THEN369
363 PRINT"ERREUR":GOTO350
364 II=1:IFLL(A)<1GOTO363
365 Y$=Z$(LL(A)):IFLL(A)<7THENY$=MID$(Y$,4,1)
366 LL(A)=LL(A)-1:GOTO410
369 II=2:IFKK(A)>12GOTO363
370 Y$=Z$(KK(A)):IFKK(A)<7THENY$=MID$(Y$,4,1)
372 KK(A)=KK(A)+1:GOTO410
400 II=3:PRINT"■■QUELLE COMBINAISON CHOISISSEZ-VOUS":INPUTY$
410 FORG=1TO6:IFY$=MID$(Z$(G),4,1)GOTO440
415 NEXT
```

```
420 FORG=7TO12:IFLEFT$(Y$,3)=LEFT$(Z$(G),3)THEN440
425 NEXT
430 PRINT"COMBINAISON INCONNUE":GOTO400
440 IFT%(A,II,G)=1THENPRINT"COMBINAISON DEJA CHOISIE!":GOTO350
450 I=II:J=G:T%(A,I,J)=1:FORW=1TO5:R(W)=B(W):NEXT
460 FORV=1TO5
470 JM=1:MM=R(JM)
480 FORW=2TO5:IFR(W)<=MMTHEN500
490 JM=W:MM=R(W)
500 NEXTW
510 PO(V)=JM:R(JM)=-1
520 NEXTV
540 ONJGOSUB3300,3300,3300,3300,3300,3300,3410,3500,3610,3710,3810,3910
580 NEXTA
585 NEXTJJ
599 FORA=1TOM:GOSUB3000
600 VV=CZ(A,1)+CZ(A,2)+CZ(A,3):SM=1032:MX=INT(100*VV/SM+.5)
610 PRINT"▨SCORE:"VV"MAXI.:"SM"TAUX:"STR$(MX)+"%▩":GOSUB5000
615 NEXT
618 PRINT"VOULEZ-VOUS JOUER UNE AUTRE PARTIE ?"
619 GETA$:IFA$<>"N"ANDA$<>"O"THEN619
620 IFA$="O"GOTO255
630 END
2800 PRINT"▯JOUEUR "M$(A):PRINT:PRINT"COUP NUMERO"JJ:PRINT
2810 PRINT"LANCER NUMERO"K"▩▩▩"
2820 FORD=1TOKM:PRINT"▮"X$(B(D))"▯▯";:NEXTD:PRINT"▩▩":RETURN
2900 PRINT"VOULEZ-VOUS VOIR LA MARQUE ?(O/N)"
2902 GETQ$
2905 IFQ$<>"O"ANDQ$<>"N"THEN2902
2910 IFQ$="N"THENRETURN
3000 PRINT"▯┌──────┬──────┬──────┬──────┐";
3005 MM$(A)="▨"+LEFT$(M$(A),5)+"▩"
3010 PRINT"| "MM$(A)" ▮MON↑ ▯DES/ | LIB |";
3020 PRINT"└──────┴──────┴──────┴──────┘";
3032 FORG=1TO6
3040 PRINT"| "+LEFT$(Z$(G),5)+" |";:FORI=1TON:OQ=C%(A,I,G):GOSUB6000:NEXTI
3045 NEXTG
3046 FORH=1TO22:PRINT"─";:NEXT
3050 FORI=1TON:CS(I)=0:FORG=1TO6:CS(I)=CS(I)+C%(A,I,G):NEXTG
3056 NEXTI
3057 G=13
3060 PRINT"| TOT 1 |";:FORI=1TON:OQ=CS(I):GOSUB6000:NEXTI
3065 FORI=1TON:T(I)=0:IFCS(I)>60THENT(I)=30
3068 NEXTI
3070 PRINT"| BONUS |";:FORI=1TON:OQ=T(I):GOSUB6000:NEXTI
3075 FORI=1TON:CT(I)=CS(I)+T(I):NEXTI
3080 PRINT"| TOT 2 |";:FORI=1TON:OQ=CT(I):GOSUB6000:NEXTI
3100 FORG=7TO12
3120 PRINT"| "+LEFT$(Z$(G),5)+" |";:FORI=1TON:OQ=C%(A,I,G):GOSUB6000:NEXTI
3130 NEXTG
3150 FORI=1TON
3155 CX(I)=0
3160 FORG=7TO12:CX(I)=CX(I)+C%(A,I,G):NEXTG
3170 CZ(A,I)=CT(I)+CX(I)
3178 NEXTI
```





```
3180 PRINT"| TOT 3 |";:FORI=1TON:OQ=CZ(A,I):GOSUB6000:NEXTI
3190 PRINT"L____|____|____|____|____J":IFJJ<36THENGOSUB5000
3195 IFJJ<37THENGOSUB2800
3200 RETURN
3300 FORV=1TO5:IFB(PO(V))=JTHENC%(A,I,J)=C%(A,I,J)+B(PO(V))
3315 NEXTV
3320 RETURN
3410 IFB(PO(1))=B(PO(2))ANDB(PO(2))=B(PO(3))ANDB(PO(4))=B(PO(5))THENC%(A,I,J)=30
3420 IFB(PO(1))=B(PO(2))ANDB(PO(3))=B(PO(4))ANDB(PO(4))=B(PO(5))THENC%(A,I,J)=30
3430 RETURN
3500 IFB(PO(1))=B(PO(2))ANDB(PO(2))=B(PO(3))ANDB(PO(3))=B(PO(4))THENC%(A,I,J)=30
3510 IFB(PO(2))=B(PO(3))ANDB(PO(3))=B(PO(4))ANDB(PO(4))=B(PO(5))THENC%(A,I,J)=30
3530 RETURN
3610 FORV=1TO4:IFB(PO(V+1))=B(PO(V))-1THENNEXT
3615 IFV=5THENC%(A,I,J)=40
3620 RETURN
3710 FORV=1TO4:IFB(PO(V+1))=B(PO(V))THENNEXTV:IFV=5THENC%(A,I,J)=50
3720 RETURN
3810 FORV=1TO5:C%(A,I,J)=C%(A,I,J)+B(PO(V)):NEXT
3820 IFT%(A,I,12)=1ANDC%(A,I,12)>=C%(A,I,J)THENC%(A,I,J)=0
3830 RETURN
3910 FORV=1TO5:C%(A,I,J)=C%(A,I,J)+B(PO(V)):NEXT
3920 IFT%(A,I,11)=1ANDC%(A,I,11)<=C%(A,I,J)THENC%(A,I,J)=0
3930 RETURN
5000 GETA$:IFA$=""THEN5000
5010 RETURN
6000 IFG=13ANDOQ=0THENPRINT"     |";:GOTO6020
6002 IFG=13GOTO6010
6005 IFT%(A,I,G)=0THENPRINT"     |";:GOTO6020
6010 PRINTSPC(4-LEN(STR$(OQ)))STR$(OQ)" |";
6020 RETURN
```

La fonction DRAW pour VIC et 64

Une amélioration substancielle de vos graphiques haute résolution

Dans "La découverte du VIC" et "La découverte du C64" (Ed. du P.S.I), la section consacrée au trace de courbes haute résolution fournit le moyen d'obtenir des graphiques spectaculaires. Ils ont cependant un défaut (cf.photo 1) : le tracé obtenu est pointillé. C'est dû au fait qu'on n'a utilisé que l'opération de tracé qu'on pourrait symboliser par "PLOT X,Y" - noircissement du point de coordonnées X,Y.

Sur ATARI et ORIC (et aussi sur VIC muni de Super-Expander) on dispose d'une fonction DRAW capable de tracer le segment de droite joignant les points X0, Y0 à X1, Y1. On obtient alors des tracés continus.

Nous allons fournir une routine BASIC qui permette d'obtenir le même résultat sur C64 ou sur VIC sans Super Expander.

Cas particulier

Nous partons du programme D-1 de "La découverte du VIC" p. 123 (p.125 pour le 64).

Pourquoi les tracés sont-ils pointillés ? Examinons la boucle centrale du tracé; elle est de la forme :

```
FOR  X = XD TO   XA (par pas de 1)
Y correspondant = ...
tracé du point X, Y
NEXT
```

On a des pointillés car on trace des points qui, s'ils correspondent bien à des X consécutifs, ne correspondent pas à des Y consécutifs. Si la pente de la fonction est élevée autour de X alors la différence Y (X+1) - Y (X) est très supérieur à 1 et les deux points sont écartés.

Dès lors, la solution est simple. Au lieu de ne tracer que les points X,Y(X) et X+1,Y(X+1) : on tracera tous les points : X,Y(X) ; X,Y(X)+1 ; ... X+1,Y(X+1)-1 ; X+1,Y(X+1).

A l'intérieur de la boucle sur X, s'ajoute maintenant une boucle sur Y. Notons XF le X+1; XP est le X précédent. Le programme devient :

```
tracé du point XD (tracé initial)
XP =XD : YP = ... (initialisations)
FOR XF = XD+1 TO XA (la boucle ex-
  terne part maintenant du 2e
  point)
YF = FNP (XF)
FOR Y = YP TO YF (boucle interne)
X = XP + (Y - YP) / (YF - YP)
  (calcul du X intermédiaire par
  interpolation)
tracé du point X,Y
NEXT Y
YP = YF : XP = XF : NEXT XF
```

Le programme ci-dessous est le programme D - 1 de "La découverte du 64" modifié selon le schéma ci-dessus.

Les instructions modifiées ou ajoutées sont : 75, 80, 85, 90, 95 et 105. La version VIC doit recevoir exactement les mêmes modifications sauf que 80 doit se lire :

```
80 : FOR XF = 1 TO 170
```

Le STEP en 90 est curieux. Il est nécessaire pour tenir compte du fait que YF peut être supérieur (le pas sera alors de 1) ou inférieur à YF (le pas sera alors -1).

Le résultat obtenu est montré photo 2 (sur 64).

```
0 REM
1 REM   PROGRAMME D-1  MODIFIE
2 REM
10 POKE53272,PEEK(53272) OR 8
20 POKE52,32:POKE56,32:CLR
30 E=8192:CO=1024
40 FOR I=CO TO CO+999:POKE I,1:NEXT
50 FOR I=E TO E+7999:POKEI,0:NEXT
60 POKE 53265,PEEK(53265) OR 32
70 DEF FNF(X)=100+90*SIN(8*π*X/318)
75 X=0:Y=100:XP=0:YP=100:GOSUB900
80 FOR XF=1 TO 318           :GOSUB1000
85 YF=FNF(XF)
90 FOR Y=YP TO YF STEP SGN(YF-YP)
95 X=XP+(Y-YP)/(YF-YP)
100 GOSUB 900:GOSUB 1000:NEXT
105 YP=YF:XP=XF:NEXT
110 GOTO 110              *YM+8*XM:RETURN
900 XM=INT(X/8):YM=INT(Y/8):P=E+320
1000 K=X AND 7:L=Y AND 7:Q=P+L
1010 POKE Q,PEEK(Q) OR 2↑(7-K):RETURN
```

Cas général

La modification ci-dessus résout le problème pour toute courbe de la forme Y = F (X). En revanche, pour une courbe de la forme X=F(T); Y=G(T) (courbe paramétrique ou cas de la résolution d'une équation différentielle), le problème n'est pas résolu car l'écart peut être supérieur à 1 à la fois en X et en Y.

Essayons donc de tracer le segment de droite X0, Y0 - X1, Y1. Une solution évidente apparaît: on applique ce qui précède à l'équation :

Y = X0 + (X-X0) * (Y1-Y0) / (X1-X0)

Mais alors on aura deux boucles imbriquées :

boucle sur les X consécutifs
boucle sur les Y comme ci-dessus.

On peut gagner une boucle en remarquant que si:

|X1-X0| > |Y1-Y0|,

les Y correspondant à deux X consécutifs seront a fortiori consécutifs. On pourra donc ne mettre que la boucle sur X , en somme comme dans le programme D-1 primitif.

Oui, mais si |Y1-Y0| < |X1-X0| ?

Très simple ! YAKA échanger les rôles de X et Y; d'où le programme ci-dessous (On a mis une instruction d'essai en 10 qui demande les coordonnées de deux points pour tracer le segment qui les joint).

Pour le VIC, les instructions 5, 20, 30, 40, 50 et 60 doivent être remplacées respectivement par les instructions 20, 10, 30, 40, 50 et 60 du programme "D-1" de "La découverte du VIC" (p.123). De même pour 900 et 1000 - sqq.

L'ensemble 70-180 peut facilement être transformé en sous-programme.

Daniel-Jean DAVID

```
0 REM
1 REM   ROUTINE DRAW POUR 64
2 REM
5 POKE52,32:POKE56,32:CLR
10 INPUT"X0,Y0,X1,Y1";X0,Y0,X1,Y1
20 POKE53272,PEEK(53272) OR 8
30 E=8192:CO=1024
40 FOR I=CO TO CO+999:POKE I,1:NEXT
50 FOR I=E TO E+7999:POKEI,0:NEXT
60 POKE 53265,PEEK(53265) OR 32
70 IF ABS(Y1-Y0) > ABS(X1-X0) GOTO 150
80 FOR X=X0 TO X1 STEP SGN(X1-X0)
90 Y=Y0+(X-X0)*(Y1-Y0)/(X1-X0)
100 GOSUB 900:GOSUB 1000:NEXT
110 GOTO 110
150 FOR Y=Y0 TO Y1 STEP SGN(Y1-Y0)
160 X=X0+(Y-Y0)*(X1-X0)/(Y1-Y0)
170 GOSUB 900:GOSUB 1000:NEXT
180 GOTO 180
900 XM=INT(X/8):YM=INT(Y/8):P=E+320
                    *YM+8*XM:RETURN
1000 K=X AND 7:L=Y AND 7:Q=P+L
1010 POKE Q,PEEK(Q) OR 2↑(7-K):RETURN
```

Photo 1

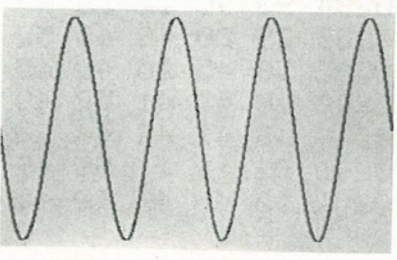


Photo 2

Auto-RUN VIC

Une astuce très demandée

Après un chargement par: LOAD "",1,1 le programme prélude se charge à partir du sommet de la pile en 01F8 en y plaçant l'adresse de la routine à exécuter (débutant en 0268 ici). Celle-ci décode une autre routine comprise entre 0200 et 0233 et lui passe le contrôle.

Celle-ci empêche alors l'action des touches RUN et RESTORE qui devront être de nouveau autorisées par le programme principal. Au passage, on modifie le vecteur "SAVE" et on place LOAD/RUN dans le tampon clavier pour permettre au programme principal qui suit de se charger et de s'exécuter automatiquement. On place également un message d'attente sur l'écran. Pour empêcher les petits malins de se passer du prélude et de charger directement le programme principal, on place une adresse "départ" en (0,1). Dans le programme principal, on fera par exemple SYS "faux départ" et "faux départ" : JMP(00).

On vérifie ainsi que le prélude a été chargé et comme il est très difficile à sauvegarder, on a en prime une excellente méthode de protection.

Je viens de me rendre compte que, si une sauvegarde était bel et bien impossible en BASIC, car on écrase la routine en utilisant le tampon BASIC, il n'en est pas de même avec le moniteur. Il faudrait pour cela placer la routine de décodage à l'intérieur même du tampon clavier qui serait alors modifié par .S . Pour la véritable sauvegarde, il suffit de faire appel à une routine extérieure qui place le code dans le tampon clavier et effectue une sauvegarde dans la foulée.

tout repose sur ces deux octets.

```
. :01F8 67 02 E9 C7 79
. :01FD C6 80 E3[D4 81
. :0202 C6 8F 48 D4 B9
. :0207 46 94 01 D4 8B
. :020C 46 99 01 D4 FE
. :0211 C6 99 81 D4 C1
. :0216 C6 BB 81 54 80
. :021B C2 E3 10 2F F2
. :0220 D4 D5 42 80 54
. :0225 DD C2 80 D4 8D
. :022A C6 87 C8 54 9A
. :022F 50 01 26 0F 65]
. :0234 93 9C 12 28 43
. :0239 29 20 48 45 52
. :023E 56 45 20 4C 45
. :0243 20 4D 41 52 43
. :0248 48 41 4E 44 20
. :024D 92 11 11 55 4E
. :0252 20 4D 4F 4D 45
```

Voici les octets chargés par le programme ECODE entre 01F8 et 0273.

Quand le chargement est achevé, le processeur donne le contrôle à la routine d'adresse (01F8) + 1 , 0268 dans l'exemple .

Ce n'est pas le cas quand le chargement est effectué par le moniteur et on peut analyser ces octets,

mais ...

```
.:0257 4E 54 2C 20 4A
.:025C 45 20 43 48 41
.:0261 52 47 45 2E 2E
.:0266 05 00 A2 33 38
.:026B 3E 00 02 CA 10
.:0270 FA 00 8D 00 00
.:0275 00 00 00 00 00
```

un désassemblage à partir de $0200 donne ceci

Des nombres aléatoires a priori.

Qui penserait que quelque chose se cache ici?

```
., 0200 ???
., 0201 STA ($C6,X)
., 0203 ???
., 0204 PHA
., 0205 ???
., 0206 LDA $9446,Y
., 0209 ORA ($D4,X)
., 020B ???
., 020C LSR $99
., 020E ORA ($D4,X)
., 0210 INC $99C6,X
., 0213 STA ($D4,X)
., 0215 CMP ($C6,X)
., 0217 ???
., 0218 STA ($54,X)
., 021A ???
., 021B ???
., 021C ???
., 021D BPL $024E
., 021F ???
., 0220 ???
., 0221 CMP $42,X
., 0223 ???
., 0224 ???
., 0225 CMP $80C2,X
., 0228 ???
., 0229 STA $87C6
., 022C INY
., 022D ???
., 022E TXS
., 022F BVC $0232
., 0231 ROL $0F
., 0233 ADC $93
```

```
., 0268 LDX #$33 (longueur
., 026A SEC        routine)
., 026B ROL $0200,X
., 026E DEX
., 026F BPL $026B
., 0271 BMI $0200
```

Dans un chargement normal, la routine ci-contre est exécutée. On obtient le véritable code et le contrôle lui est ensuite donné par BMI $0200. (Voir ci-après)

```
.' 0234[illegible]
.' 0240[illegible]
.' 024C[illegible]
.' 0258[illegible]
.' 0264[illegible]"38>..J.
```

P.S: avant de sauvegarder ce prélude, il faut coder la routine par l'opération inverse :

```
LDY #$33
LDX #$00
ROR $0200,X
INX
DEY
BPL
BRK
```

Remarque : l'octet 026A est SEC car le dernier octet est CB (en 0233) qui a son bit 0 égal à 1.

```
.,  0200 LDA #$03         ; empêche RESTORE
.,  0202 STA $911E
.,  0205 LDA #$72         ; empêche STOP
.,  0207 STA $0328
.,  020A LDA #$22         ; réinitialise si on tape SAVE
.,  020C STA $0332
.,  020F LDA #$FD
.,  0211 STA $0333
.,  0214 LDA #$83         ; met LOAD/RUN dans le tampon
.,  0216 STA $0277          clavier
.,  0219 LDA #$01
.,  021B STA $C6
.,  021D JSR $E55F        ; efface l'écran
.,  0220 LDA #$AA         ; pour s'assurer que l'utilisa-
.,  0222 STA $00          teur charge ce prélude on range une
.,  0224 LDA #$BB         adresse BBAA en (0,1). Dans le pro-
.,  0226 STA $01          gramme principal, on fera quelque
.,  0228 LDA #$1B         part un JMP ($00).
.,  022A STA $900F        ; Fond blanc, bord bleu.
.,  022D LDA #$34
.,  022F LDY #$02         ; Imprime un texte (nom de l'au-
.,  0231 JMP $CB1E        teur, message d'attente), puis,
                          grâce au tampon, il y aura LOAD
                          puis RUN.
```

Hervé LE MARCHAND

Adresses Stratégiques du 64

HEXA	DEC	DESCRIPTION
0000	0	registre direction du port parallèle dans le 6510
0001	1	registre donnée de ce port
0002	2	inutilisé
0003-0004	3-4	vecteur pour conversion réel-entier
0005-0006	5-6	vecteur pour conversion entier-réel
0007	7	caractère recherché
0008	8	indicateur mode guillemets
0009	9	sauvegarde colonne de tabulation
000	10	0=LOAD 1=VERIFY
000B	11	pointeur dans tampon d entrée/numéro d'indice
000C	12	indicateur de DIM par défaut
000D	13	type : $FF = chaîne 00 = numérique
000E	14	type : $80 = entier 00 = réel
000F	15	indicateur mémoire mode guillemets en LIST, balayage DATA
0010	16	indicateur indice / FNx
0011	17	0=INPUT, $40 = GET, $98 = READ
0012	18	signe pour TAN/indicateur de comparaisons
0013	19	indicateur actuel d'interrogation d'entrée
0014-0015	20-21	valeur entière
0016	22	rangement temporaire du pointeur de pile
0017-0018	23-24	dernier vecteur de chaîne temporaire
0019-0021	25-33	pile pour chaînes temporaires
0022-0025	34-37	zone de pointeurs utilitaires
0026-002A	38-42	zone pour le produit dans une multiplication
002B-002C	43-44	pointeur : début du texte BASIC
002D-002E	45-46	pointeur : début des variables
002F-0030	47-48	pointeur : début des tableaux
0031-0032	49-50	pointeur : fin des tableaux
0033-0034	51-52	pointeur : zone des chaînes
0035-0036	53-54	pointeur utilitaire chaînes
0037-0038	55-56	pointeur : fin de mémoire
0039-003A	57-58	numéro de ligne BASIC courante
003B-003C	59-60	numéro de ligne BASIC précédente
003D-003E	61-62	pointeur instruction BASIC pour CONT
003F-0040	63-64	numéro de ligne DATA courante
0041-0042	65-66	adresse courante de DATA
0043-0044	67-68	vecteur pour INPUT
0045-0046	69-70	nom de la variable courante
0047-0048	71-72	adresse de la variable courante
0049-004A	73-74	pointeur vers variable pour FOR...NEXT
004B-004C	75-76	sauvegarde de Y ou d opérateur ou du pointeur BASIC

HEXA	DEC	DESCRIPTION
004D	77	accumulateur des symboles de comparaison
004E-0053	78-83	zone de travail diverse
0054-0056	84-86	vecteur de saut pour fonctions
0057-0060	87-96	zone de travail numérique diverse
0061	97	Acc. numéro 1 : exposant
0062-0065	98-101	Acc. numéro 1 : mantisse
0066	102	Acc. numéro 1 : signe
0067	103	pointeur vers constantes pour évaluer une série
0068	104	débordement de Acc. numéro 1
0069-006E	105-110	Acc. numéro 2 : exposant mantisse signe
006F	111	signe de la comparaison Acc.1 Acc.2
0070	112	arrondi de l'Acc. numéro 1
0071-0072	113-114	longueur tampon-cassette / pointeur série
0073-008A	115-138	sous-programme CHRGET (obtient un caractère Basic)
007A-007B	122-123	pointeur Basic (dans CHRGET)
008B-008F	139-143	racine pour RND
0090	144	mot d'état ST
0091	145	copie du PIA clavier : indicateurs de STOP et RVS
0092	146	constante de vitesse pour cassette
0093	147	0 = LOAD ; 1 = VERIFY
0094	148	sortie série : indicateur de caractère différé
0095	149	caractère série différé
0096	150	synchro cassette
0097	151	sauvegarde de registre
0098	152	nombre de fichiers ouverts
0099	153	périphérique standard d'entrée (normalement 0)
009A	154	périphérique de sortie standard déterminé par CMD (normalement 3)
009B	155	parité du caractère sur cassette
009C	156	indicateur de caractère reçu
009D	157	contrôle de sortie : directe = $80 RUN = 0
009E	158	témoin d'erreur sur cassette passe 1/tampon caractère
009F	159	témoin d'erreur corrigée sur cassette (passe 2)
00A0-00A2	160-162	horloge au 60e de seconde
00A3	163	compte de bits série / indicateur d EOI
00A4	164	compteur de cycles
00A5	165	décompteur écriture cassette / compteur de bits
00A6	166	pointeur dans tampon cassette
00A7	167	compteur en-tête écriture cassette / passe lecture/ bit entrée
00A8	168	nouvel octet écriture cassette / erreur lecture / compte bits entrées
00A9	169	écrit bit départ/ erreur lecture bit / bit de départ
00AA	170	balayage cassette / assemblage octet
00AB	171	longueur en-tête écriture / checksum lecture / parité
00AC	172-173	pointeur : tampon cassette déroulement écran
00AE-00AF	174-175	adresse fin cassette / fin du programme
00B0-00B1	176-177	constantes de vitesse pour cassette
00B2-00B3	178-179	pointeur : début du tampon cassette
00B4	180	1 = temporisateur cassette activé /compte de bits
00B5	181	EOT cassette / prochain bit à envoyer en RS232
00B6	182	erreur caractère en lecture / tampon octet à sortir
00B7	183	nombre de caractères du nom de fichier
00B8	184	fichier logique courant
00B9	185	adresse secondaire courante
00BA	186	périphérique courant
00BB-00BC	187-188	pointeur vers nom de fichier
00BD	189	mot à décaler en sortie / caractère entré
00BE	190	nombre de blocs restant à lire ou écrire
00BF	191	tampon mot série

HEXA	DEC	DESCRIPTION
00C0	192	commande moteur cassette
00C1-00C2	193-194	adresse départ d E/S
00C3-00C4	195-196	pointeur routine de préparation du noyau
00C5	197	dernière touche appuyée
00C6	198	nombre de caractères dans le tampon clavier
00C7	199	indicateur de contraste inversé
00C8	200	pointeur vers fin de ligne en entrée
00C9-00CA	201-202	témoin curseur en entrée (ligne/colonne)
00CB	203	touche enfoncée : 64 si aucune
00CC	204	0 = le curseur clignote
00CD	205	décompteur pour vitesse de clignotement curseur
00CE	206	caractère sous le curseur
00CF	207	phase de clignotement curseur
00D0	208	entrée depuis l'écran ou le clavier
00D1-00D2	209-210	pointeur vers ligne écran
00D3	211	position du curseur dans la ligne
00D4	212	0 = curseur direct ; sinon programmé
00D5	213	longueur ligne courante sur écran
00D6	214	ligne où se trouve le curseur
00D7	215	tampon / checksum / dernière touche
00D8	216	nombre d'insertions en attente
00D9-00F0	217-240	table des liens des lignes-écran
00F1	241	lien bidon
00F2	242	marqueur ligne écran
00F3-00F4	243-244	pointeur mémoire couleur écran
00F5-00F6	245-246	pointeur table décodage clavier
00F7-00F8	247-248	pointeur vers réception RS232
00F9-00FA	249-250	pointeur vers émission RS232
00FB-00FE	251-254	libre
00FF	255	donnée temporaire BASIC
0100-01FF	256-511	pile processeur dont sont utilisés temporairement :
00FF-010A	255-266	zone de travail pour la conversion réel ASCII
0100-013E	256-318	mémorisation erreurs cassette
0200-0258	512-600	tampon d entrée du Basic
0259-0262	601-610	table des fichiers logiques
0263-026C	611-620	table des numéros de périphériques
026D-0276	621-630	table des adresses secondaires
0277-0280	631-640	tampon clavier
0281-0282	641-642	plus petite adresse de RAM Basic
0283-0284	643-644	plus grande adresse de RAM Basic
0285	645	indicateur de time-out sur le bus série
0286	646	code couleur courant
0287	647	couleur sous le curseur
0288	648	page de la mémoire d'écran
0289	649	taille maximum du tampon d'écran (doit être < 10)
028A	650	répétition automatique des touches : 0 = touches curseur ; 127 = aucune touche 128=toutes les touches
028B	651	compteur pour la vitesse de répétition
028C	652	compteur du délai pour la répétition
028D	653	indicateur de SHIFT C= CTRL : 0=rien 1=SHIFT 2=C= 4= CTRL
028E	654	dernier motif de SHIFT
028F-0290	655-656	pointeur vers la routine d'établissement de la table de transcodage clavier
0291	657	mode des touches. $00: désactive les touches "SHIFT"; $80: les active
0292	658	0 = déroulement écran autorisé
0293	659	contrôle du RS232
0294	660	commande du RS232
0295-0296	661-662	temps d un bit pour le RS232

HEXA	DEC	DESCRIPTION
0297	663	état du RS232
0298	664	nombre de bits à envoyer
0299-029A	665-666	code vitesse du RS232
029B	667	pointeur réception RS232
029C	668	pointeur entrée RS232
029D	669	pointeur émission RS232
029E	670	pointeur sortie RS232
029F-02A0	671-672	sauvegarde IRQ durant E/S cassette
02A1	673	active RS232
02A2	674	horloge TR pendant opération cassette
02A3	675	donnée temporaire pendant lecture cassette
02A4	676	indicateur IRQ pendant lecture cassette
02A5	677	index de ligne temporaire
02A6	678	0=NTSC 1=PAL
02A7-02FF	679-767	inutilisé
0300-0301	768-769	pointeur message d'erreur
0302-0303	770-771	pointeur vers reset tiède de Basic
0304-0305	772-773	pointeur vers routine de codage des mots-clés Basic
0306-0307	774-775	pointeur vers routine d'impression des codes des mots-clés
0308-0309	776-777	pointeur vers départ nouvelle ligne Basic
030A-030B	778-779	pointeur vers routine d'évaluation des codes BASIC
030C-030F	780-783	zone à partir de laquelle sont chargés A.X.Y et P lors d'un SYS.
0310	784	instruction de saut ($4C) pour USR
0311-0312	785-786	pointeur vers adresse d'USR (B248)
0313	787	libre
0314-0315	788-789	vecteur interruption hardware ($EA31)
0316-0317	790-791	vecteur de BRK (FE66)
0318-0319	792-793	vecteur de NMI (FE47)
031A-031B	794-795	vecteur pour OPEN (F34A)
031C-031D	796-797	vecteur pour CLOSE (F291)
031E-031F	798-799	vecteur pour établir l'entrée standard (F20E)
0320-0321	800-801	vecteur pour établir la sortie standard (F250)
0322-0323	802-803	vecteur pour restaurer les E/S standard (F333)
0324-0325	804-805	vecteur pour INPUT (F157)
0326-0327	806-807	vecteur pour sortie (F1CA)
0328-0329	808-809	vecteur pour tester STOP (F6ED)
032A-032B	810-811	vecteur pour GET (F13E)
032C-032D	812-813	vecteur pour abandonner les E/S (F32F)
032E-032F	814-815	vecteur pour USR (FE66)
0330-0331	816-817	pointeur pour LOAD (F4A5)
0332-0333	818-819	pointeur pour SAVE (F5ED)
033C-03FB	828-1019	tampon cassette
0400-07FF	1024-2047	mémoire d écran
0800-9FFF	2048-40959	texte BASIC
8000-9FFF	32768-40959	(option) ROM en cartouche
A000-BFFF	40960-49151	RAM ou ROM BASIC
C000-CFFF	49152-53247	RAM
D000-D02E	53248-53294	boîtier V.I.C. 6567
D400-D41C	54272-54300	boîtier SID 6581 (synthétiseur)
D800-DBFF	55296-56319	RAM couleur (quartets)
DC00-DC0F	56320-56335	6526 n°1
DD00-DD0F	56576-56591	6526 n°2
D000-DFFF	53248-57343	générateur de caractères
E000-FFFF	57344-65535	RAM ou ROM noyau du S.E.

Daniel-Jean DAVID

Atari - Trucs

Affichage clignotant

Ce petit sous-programme fait clignoter un message d'interrogation. Les POKE "magiques" sont :

752: 0= curseur actif
     1= curseur inhibé
656: ligne où est le curseur
764: touche actuellement enfoncée

```
1000 POKE 752,1
1010 DIM A$(22), B$(22), X$(22)
1020 A$ ="APPUYEZ SUR UNE TOUCHE"
1030 B$ = "ESC CLEAR"
1040 X$ = A$
1050 FOR I = 1 TO 100
1060 POKE 656,2 : PRINT X$
1070 FOR I = 1 TO 300 : NEXT J
1080 IF PEEK(764) = 255 THEN 1100
1090 RETURN
1100 IF X$ = A$ THEN X$ = B$
:GOTO 1120
1110 X$ = A$
1120 RETURN
```

La séquence d'appel sera, par exemple :

```
100 GOSUB 1000 : POKE 764, 0 :
PRINT B$
```

Désactivation de la touche BREAK

Il est quelquefois gênant de risquer de sortir d'un programme par appui accidentel sur la touche "BREAK". On la désactive par :

POKE 16,64 : POKE 53774,64

On la réactive par :

POKE 16,192 : POKE 53774,192

Pour la touche RESET, c'est différent. Il y a en 12,13 un vecteur qui pointe vers la routine qui sera exécutée si on appuie sur RESET. Donc, pour changer le comportement de la touche RESET, il faut écrire une routine en langage machine qui effectue les opérations de RESET souhaitées, et mettre son adresse de départ en 12 (partie basse) et 13 (partie haute).

Rita A...

Imprimer en langage machine

Une des difficultés du langage machine est que l'on doit tout faire, même pour les opérations les plus simples. Parmi celles-ci, une opération dont on a un besoin universel est d'imprimer sur l'écran. On peut s'en tirer par un

```
LDA code écran
STA adresse voulue de la
        mémoire d'écran
```

Mais il faut gérer l'adresse écran en fonction de la position où l'on veut imprimer.

Pourquoi ne pas utiliser la routine système qui le fait ? Pour imprimer un caractère à la position du curseur sur l'écran, il suffit de faire:

```
LDA  code ASCII du caractère
JSR $F6A4
```

En somme, F6A4 est à l'ATARI ce que FF D2 est aux COMMODORE (sur ORIC c'est CC12)

Daniel-Jean DAVID

Cours de Langage-Machine
I

Comme promis, nous commençons à ce numéro notre cours de langage machine. Bien sûr, il est commun à toutes les machines traitées dans La Commode : les CBM, VIC et 64, ORIC et ATARI.

Pourquoi programmer en langage-machine ?

Comme nous allons le voir, la programmation en langage machine est plus délicate qu'en BASIC. Alors, est-ce par masochisme que vous lisez ces lignes ?

Il y a peut-être un peu de cela chez ceux pour qui la programmation est un sport. Pour ceux-là, une fois aguerris aux pièges de BASIC, un moyen de se mesurer à des obstacles plus sérieux et d'élargir leurs horizons est de se battre avec le langage machine.

Mais il y a d'autres raisons plus vitales :

1 - Le langage machine permet d'accéder à des ressources de la machine inaccessible aux langages évolués.

2 - Le langage machine augmente considérablement les performances d'un programme, tant en taille mémoire occupée qu'en vitesse de traitement.

<u>1</u> - Par définition - et c'est pour simplifier la tâche du programmeur- un langage évolué manipule les données sous forme de variables, connues uniquement par leur nom. En langage machine, on manipule directement les cases mémoire par leur adresse; de plus, on peut manipuler directement le contenu des registres de calcul du processeur.

Un autre élément inacessible à BASIC est la possibilité de définir votre propre routine d'interruption ce qui a des applications très intéressantes.

<u>2</u> - Comme vous le savez, l'utilisation de BASIC passe par une étape de traduction de votre programme en langage machine, car en dernier ressort, la machine ne peut exécuter que du langage machine. Cette traduction est effectuée par un programme, le compilateur ou l'interpréteur BASIC. Pour ne pas être trop complexe, celui-ci doit, dans certains cas, avoir une attitude stéréotypée, négligeant les cas particuliers.

<u>Ex.</u> Calcul de x^y :
On effectue en fait $e^{y \text{ Log } x}$. Dans le cas où y = 2, on pourrait effectuer x*x, mais, le plus souvent, le compilateur restera dans le cas général et calculera x^2 en prenant le même temps que pour $x^{2,1}$.

Un interpréteur perd encore plus de temps qu'un compilateur. Un programme en BASIC interprété va de 10 à 100 fois plus lentement que l'équivalent écrit par un bon programmeur en langage machine.

Ce facteur peut être vital pour certaines applications d'où la nécessité de programmer en langage machine dans certains cas.

Pour vous persuader de la différence de vitesse entre langage machine et BASIC, faites l'expérience suivante: comparez la durée

de vidage de l'écran obtenue par :

1) PRINT "CLR" (ATARI : ESC CLEAR, ORIC : CTRL L)

et

2) FOR I = E TO E + X : POKE I, B : NEXT

avec:

modèle	E	X	B
CBM	32768	999	32
VIC	256*PEEK (648)	505	32
64	1074	999	32
ORIC	48000	999	32
ATARI	PEEK(88)+ 256*PEEK(89)	999	0

La solution 1 appelle la routine langage machine du système qui fait exactement la même chose que 2. Convaincus ? Eh bien, le langage machine procure aussi des gains du point de vue occupation mémoire. Ex. supposons que vous vouliez manipuler un tableau de nombres. Eh bien, au mieux, BASIC vous permettra de constituer un tableau d'entiers soit 2 octets par nombre. Maintenant, il se peut que vous sachiez a priori que vos nombres sont tous inférieurs à 256. En langage machine, vous n'occuperez qu'un octet par nombre, d'où gain d'un facteur 2. Là encore cela peut être vital.

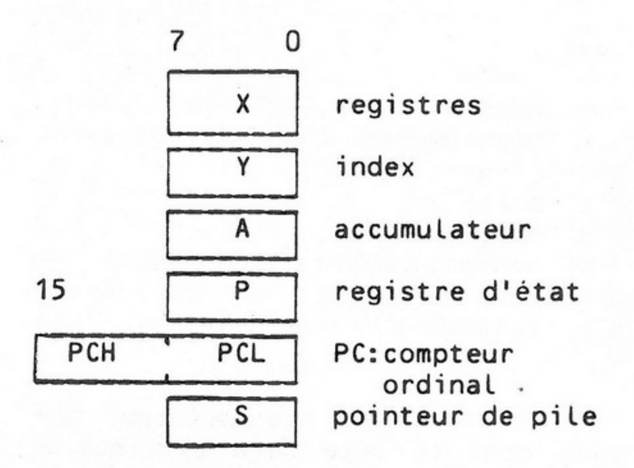
Comment se présente le langage-machine

. Informations Manipulées

Le langage-machine ne connaît que deux types d'objets manipulables : les registres internes du processeur et les cases mémoire.

Les cases mémoires sont manipulées une par une (octet par octet), par leur adresse, avec le microprocesseur 6502 qui nous occupe. Si l'on veut représenter certaines données occupant plusieurs octets, on peut le faire, mais leur manipulation se fera tout de même octet par octet. Les problèmes de représentation des données seront traités plus en détail dans le prochain article de cette série.

Les registres internes du 6502 sont peu nombreux. Ils obéissent au schéma suivant :



L'accumulateur A est le principal registre de calcul. Sa largeur est de 8 bits. C'est sur lui que sont effectuées la plupart des opérations arithmétiques : la plupart des opérations sont effectuées entre l'accumulateur et une case mémoire et le résultat se retrouve dans l'accumulateur.

Le compteur ordinal PC est le seul registre 16 bits du 6502. A tout instant, il contient l'adresse de la prochaine instruction à exécuter. Il est incrémenté automatiquement par les instructions séquentielles et il reçoit une nouvelle valeur lors d'une instruction de saut ou de branchement. Il a 16 bits car l'espace adressable du 6502 est de 64 K octets.

Les registres X et Y servent à effectuer des comptages. Sinon, leur rôle principal concerne l'adressage indexé et ils seront décrits plus en détail avec les modes d'adressage dans le prochain article.

Le registre P regroupe les bits indicateurs des différents états qui résultent de conditions particulières qui peuvent avoir lieu au cours des calculs : par exemple, c est à 1 s'il vient de se produire une retenue, z est mis à 1 si une comparaison vient de déceler qu'il y a égalité, etc...
Le registre P obéit au schéma suivant:

7 6 5 4 3 2 1 0

n	v		b	d	i	z	c

avec :

n : négatif
v : débordement
bit 5 inutilisé
b : break
d : mode décimal
i : inhibition interruptions
z : zéro ou égalité
c : retenue .

Enfin S est le pointeur de pile dont le rôle sera expliqué à propos des sous-programmes.

. Jeu d'instructions

Ces registres et les cases mémoire sont manipulés par un répertoire d'instructions assez réduit. De plus, les instructions sont elles-mêmes très élémentaires alors que les instructions d'un langage évolué sont synthétiques. Nous proposons la classification suivante (pour donner une idée des possibilités du langage-machine et de l'aspect des instructions) celles-ci seront étudiées en détail dans les prochains articles) :

1 . instructions de transfert d'information sans traitement

- transfert entre registres. Ex. copier A dans X

- transfert entre mémoire et registre. Ces instructions sont très importantes puisque, comme les opérations se font dans le processeur sur les registres, pour agir sur une donnée il faut l'amener de la mémoire vers un registre (le plus souvent A)

Amener une donnée de la mémoire dans un registre s'appelle un "chargement". Envoyer une donnée d'un registre vers la mémoire s'appelle un "rangement". Ex. charger A, ranger A, charger X, etc.,

2 . Instructions arithmétiques et logiques :

- à 2 opérandes qui sont alors l'accumulateur et une case mémoire. Le résultat va dans A. Ex. ajouter le contenu de la case mémoire M à l'accumulateur, qui peut se symboliser par A -- A + M

- à 1 opérande qui peut être un registre ou une case mémoire. Ces opérations sont soit des incrémentations/décrémentations (Ex. incrémenter X), soit des décalages/rotations (Ex.décaler à gauche le contenu d'une case mémoire - cela revient à multiplier par 2)

3 . Opérations sur les indicateurs d'état :

- opérations inconditionnelles. Ex. forcer à 1 l'indicateur de mode décimal

- comparaisons qui positionnent les indicateurs n, z et C en fonction du résultat de la comparaison.

Rappelons que les opérations arithmétiques positionnent automatiquement certains indicateurs.

4 . Instructions de sauts et branchements :

- branchements en fonction de l'état d'un indicateur (Ex.branchements si négatif)

- sauts inconditionnels : sauts, appel et retour de sous-programme, etc..,

5 . Instructions d'entrées-sorties

Ces instructions sont absentes sur 6502 où, grâce aux boîtiers d'entrées-sorties comme le VIA, on peut par de simples lectures et écritures à l'adresse d'un port, prendre connaissance de ou influer sur l'état électrique des lignes correspondantes.

On le voit, les instructions

sont très élémentaires. Ainsi, l'équivalent de N = L + M exigera quatre instructions :

- forcer à zéro la retenue
- amener dans l'accumulateur le contenu de la mémoire M
- ajouter à l'accumulateur le con tenu de la mémoire L
- ranger le contenu de l'accumulateur

(c'est maintenant la somme) dans la mémoire N .

· Les deux formes de langage-machine

Il y a en fait deux formes bien différentes de programmation en langage-machine le binaire et l'assembleur symbolique.

Dans la première forme, on manipule des images exactes des octets qui forment les instructions en mémoire telles qu'elles seront exécutées. Chaque instruction comporte plusieurs octets dont le premier est appelé "code opération" car il spécifie la nature de l'opération. Les suivants éventuels (il y a au maximum trois octets pour une instruction sur le 6502) spécifient l'adresse mémoire concernée. En fait, on n'écrira pas en binaire, mais en hexadécimal (système à base 16). Le constructeur fournit la correspondance entre les instructions et leur code en hexadécimal.

Ex. : Dans notre exemple ci-dessous assignons les adresses hexadécimales 1000, 1001, 1002 respectivement à L, M et N et écrivons le programme à partir de $2000.

On aura :

```
$2000 18          (force c à 0)
$2001 AD 01 10    (charge M)
$2004 6D 00 10    (ajoute L)
$2007 8D 02 10    (range en N)
```

On remarque que les deux octets d'une adresse sont inversés dans l'instruction (Ex. 00 10 pour $1000) ce qui ne facilite pas les choses.

La première forme est bien rébarbative même si l'on utilise l'hexadécimal et non le binaire. La deuxième forme est beaucoup plus symbolique et elle constitue un véritable langage qu'on appelle langage assembleur symbolique.

Dans ce langage, toute instruction est formée de 4 champs séparés par au moins un espace :

1 - l'étiquette, symbole qui permet de repérer une instruction, par exemple pour s'y référer lors d'un saut. Ce champ est facultatif.

2 - le code opération -seul champ obligatoire- qui spécifie la nature de l'instruction par un petit mot, le plus souvent de 3 lettres qui s'appelle "mnémonique" car il rappelle la nature de l'instruction. Malheureusement, l'effet de mémorisation ne joue à plein que si on connaît l'anglais.
Ex. Le chargement de l'accumulateur se dit LDA (Load Accumulator), le saut se dit JMP (Jump), etc.,

3 - le champ opérande dans lequel les cases mémoire sont désignées par des noms comme les variables en BASIC.

4 - le champ commentaire qui permet de documenter le programme.
Ex. : notre programme d'addition ci-dessus s'écrirait alors :

```
DEBUT CLC   ; force c à 0
      LDA M
      ADC L ; addition
      STA N ; range le résultat
```

Cela prend un aspect plus humain et, de fait, toute programmation en langage-machine se fait d'abord sous cette forme. Mais il y a un autre problème : il faut traduire cette forme en binaire. Cela s'accomplit par un programme qui est en somme le compilateur de ce langage et s'appelle un "assembleur" .

La sortie imprimée fournie par un tel assembleur pourrait avoir l'aspect :

```
100 :                    *      =  $2000      ;FIXE L'ORIGINE
101 :2000 18                    CLC           ;FORCE C A 0
102 :2001 AD 01 10   DEBUT      LDA M
103 :2004 6D 00 10              ADC L         ;ADDITION
104 :2007 8D 02 10              STA N         ;RANGE LE RESULTAT
```

La première ligne du listing montre une 2éme sorte d'instructions de l'assembleur : c'est une instruction que l'assembleur ne traduit pas en hexa mais utilise pour sa traduction. On l'appelle "directive" et elle joue le même rôle que les déclarations dans certains langages évolués.

De quoi avez-vous besoin pour programmer en langage-machine ?

Aides logicielles

Il vous faut de toute évidence disposer d'aides à la programmation, pour pouvoir préparer vos programmes machine, les assembler et les entrer en mémoire. Il faut aussi pouvoir les exécuter, surveiller l'exécution puis les corriger et ainsi de suite.

Vous aurez donc besoin de deux aides logicielles au moins :

1 . le moniteur

Il permet d'entrer des octets en hexa dans la mémoire, de les modifier, de les sauver et récupérer sur mémoire de masse. Il permet d'exécuter un programme en langage-machine, éventuellement en pas-à-pas, ce qui est idéal pour la mise au point.

Si votre programme est assez court, vous pouvez en réaliser l'assemblage à main à l'aide de la table des codes fournie par le constructeur. A ce moment, le moniteur est le seul logiciel nécessaire d'autant que souvent, les moniteurs perfectionnés contiennent un mini-assembleur qui traduit les mnémoniques (mais pas les opérandes symboliques).

2 - l'assembleur symbolique

Il permet d'éditer votre programme symbolique et de le traduire en hexadécimal.

Quels sont les logiciels disponibles ?

- sur CBM vous avez la disquette assembleur qui contient le moniteur EXTRAMON (avec pas-à-pas) et l'assembleur.

De toutes façons, un mini-moniteur est incorporé en ROM. Vous pouvez l'utiliser couplé à la cassette Assembleur Universel de LA COMMODE.

- sur VIC il existe une cartouche moniteur (VICMON ou HESMON). Sur 64 la disquette démonstration contient un moniteur perfectionné. Il existe une version de l'Assembleur Universel LA COMMODE pour ces deux machines (Attention il faut un VIC 8K)

- sur ORIC nous recommandons, comme moniteur, la cassette Monitor 1.0 de LORICIELS. Comme assembleur, il y a une version de la cassette LA COMMODE.

- sur ATARI nous publierons, dans le prochain article, un mini-moniteur en BASIC. Comme assembleur, bien sûr, la cassette LA COMMODE.

les deux contextes de programmation en L.M. :

On peut programmer en langage-machine dans deux contextes :

1 - programmes totalement autonomes
2 - programmes intéragissant avec BASIC

C'est surtout le contexte 2 qui est utilisé. Il introduit deux contraintes :

- il faut faire attention à respecter les zones mémoire que BASIC se réserve

- il faut savoir comment charger le programme en L.M avec BASIC, comment l'appeler, etc.,

Bien sûr, le contexte 1 laisse plus de liberté, mais il implique que vous écriviez tout, notamment les entrées-sorties, alors que le contexte 2 permet d'utiliser celles de BASIC - nous vous apprendrons à le faire.

Bibliographie

Les notions que nous exposons ici sont explicitées du point de vue hardware dans "Systèmes à Microprocesseurs" chez EDITESTS, et du point de vue "contexte BASIC" dans la "Pratique du ..." Vol.II chez P.S.I. Le livre correspondant sur CBM est paru. Ceux sur VIC, 64, ORIC et ATARI devraient l'être avant le prochain article de cette série.

Nous ne vous recommandons pas du tout le livre de Monteil chez EYROLLES, dont le titre "Assembleur facile du 6502" est en lui-même une escroquerie. Il n'est pas possible de rendre l'assembleur facile; il s'agit de rendre les lecteurs capables d'en surmonter les difficultés. Telle est l'ambition de LA COMMODE.

Le prochain article examinera les questions :

- représentation binaire des données,
- utilisation du moniteur,
- utilisation de l'assembleur Universel LA COMMODE

et commencera l'étude systématique du jeu d'instructions.

Daniel-Jean DAVID

VIC A BRAC

De toutes les couleurs

Il n'est pas possible d'avoir plusieurs fonds d'écran de couleurs différentes simultanément? c'est à voir avec le programme suivant :

```
10 FOR I = 0 TO 8 : READ X : POKE
    7424 + I, X : NEXT : SYS 7424
20 DATA 173,4,144,141,15,144,24,
    144,247
```

Hervé LE MARCHAND

Petit chat

En exemple d'utilisation des équivalences données dans le VIC à BRAC du n° 8, voici un petit programme permettant d'afficher le petit chat de "Clefs pour le VIC":

```
10 POKE 36869,206 si 16 ou 8K
    254 si  8K
20 FOR I = 6144 TO 6151 : READ A :
    POKE I, A : NEXT
30 C = 37888 + 4 * (PEEK (36866)
    AND 128) : REM adresse
    mémoire couleur
40 PRINT CHR$ (147) : FOR I = 0 TO
    505 : POKE C + I , 5 : NEXT
50 DATA 56, 56, 16, 56, 125, 125,
    125, 62
```

La touche @ ou "a commercial" imprime le petit chat. L'instruction POKE 4096,0 (ou POKE 7680,0 s'il y a moins de 8K d'extension mémoire) le fait apparaître dans le coin supérieur gauche de l'écran.

Luc GRILLET

Chargement sur PET/CBM de programmes de VIC ou 64

Il peut être utile de charger sur PET/CBM un programme de VIC ou 64. Par exemple pour faire un listing chez un ami qui a un PET avec imprimante alors que vous n'en avez pas encore sur votre VIC.

L'ennui est que :

1) les adresses de début de BASIC ne sont pas les mêmes,

2) le PET/CBM n'a pas la "translation automatique" pour s'y adapter.

Avant de charger, faites :

```
POKE 4096,0 : POKE 41,16
    (VIC de base)
POKE 4608,0 : POKE 41,18
    (VIC > 8K)
POKE 2048,0 : POKE 41,8
    (C 64)
```

AUTRE METHODE :

Vous pouvez aussi : créer une ligne 0 REM, puis charger votre programme. Ensuite :

POKE 1026, 16 (VIC de base), POKE 1026,18 (VIC > 8K) ou POKE 1026,8 (C 64).

Enfin, supprimez la ligne 0 (par 0 RETURN)

Attente d'une touche quelconque

Deux façons :

```
10 GET A$ : IF  A$ = ""GOTO 10
    ou
10 POKE 198,0 : WAIT 198,1 :
    POKE 198,0
```

La première marche sur tout COMMODORE. La 2ème marche aussi sur 64. Pour les autres CBM, il faut remplacer 198 par 158.

Overlays

La commande LOAD peut s'employer à l'intérieur d'un programme: elle permet de charger une 2ème

partie si la mémoire est insuffisante ou elle permet l'emploi de menus.

Elle donne quelquefois des ennuis. Vous pouvez employer la séquence ci-dessous à sa place :

```
100 PRINT "Clr LOAD" + CHR$ (34)+
    "nom" + CHR$ (34)
110 PRINT "b b b RUN"
120 POKE 198,3 : POKE 632,19 :
    POKE 633,13 : POKE 634,13
```

Pour déterminer la couleur du fond et du cadre

```
10 INPUT "COULEUR DU FOND"; F$
20 INPUT "COULEUR DU CADRE"; C$
30 RESTORE : FOR I = 1 TO 16
40 READ A$ : IF A$ = F$ GOTO 70
50 NEXT
60 PRINT "ERREUR" : GOTO 10
70 RESTORE : FOR J = 1 TO 8
80 READ A$ : IF A$ = C$ GOTO 110
90 NEXT
100 PRINT "ERREUR" : GOTO 20
110 I=I-1:J=J-1
120 POKE 36879,(16*I) OR 8 OR J
200 :
221 :
230 :GOTO10
1000 DATA NOIR,BLANC,ROUGE
1010 DATA TURQUOISE,POURPRE,VERT
1020 DATA BLEU,JAUNE,ORANGE
1030 DATA ORANGE CLAIR,ROSE
1040 DATA TURQUOISE CLAIR
1050 DATA POURPRE CLAIR,VERT CLAIR
1060 DATA BLEU CLAIR,JAUNE CLAIR
```

Daniel-Jean DAVID

Désactivation de SUPER EXPANDER

SYS 64850 désactive les touches programmables
SYS 58232 désactive les autres commandes

Si l'on veut que la cartouche se comporte comme une extension RAM 3K, remplacer le second POKE par :

```
POKE 641,0 : POKE 642,4 : POKE
    643,0 :
POKE 644,30: POKE 648,30 : SYS
    64824
```

Simulation du VIC de base sur un VIC étendu

C'est en fait la meilleure solution pour adapter des programmes haute résolution écrits pour VIC de base à un VIC étendu : une fois le début de BASIC mis en $2000, on revient aux adresses écran du VIC de base.

1ère façon:

```
10 POKE 642,32 : REM CHANGE LE
   DEBUT BASIC
20 POKE 36869,240 : POKE 36866,150:
   REM REORGANISATION DE L'ECRAN
30 POKE 0, 108 : POKE 1,0 : POKE
   2,192 : SYS 0 : REM REDEMAR-
   RAGE DE BASIC
```

2ème façon :

Avec les instructions ci-dessous, on peut reconfigurer le VIC de façon à simuler n'importe quelle taille mémoire :

```
POKE 641,0 : POKE 642, X : POKE
   643,0
POKE 644,Y : POKE 648,Z : SYS
   64824
```

avec :

X (642)	Y (644)	Z (648)	mémoire
16	30	30	base
4	30	30	+3K
18	64	16	+8K
18	96	16	+16K
18	128	16	+24K

En fait, pour faire démarrer BASIC en $2000, c'est 32 qu'il faut mettre en 642 (VIC > 8K)

On peut aussi faire croire à un VIC qu'il a plus de mémoire qu'en réalité. Un plantage suit assez vite cette fantaisie.

Pierre-Etienne THALBERG

Un manche à balai pour votre ORIC

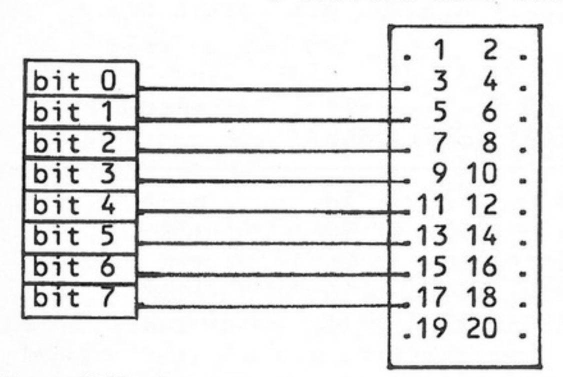
Un peu de théorie

Vous disposez sur l'ORIC d'un port parallèle utilisateur : c'est tout simplement le port imprimante.

Comment fonctionne-t-il ? Comme tout port d'entrée-sortie. Mais encore ? nous donnons ici une description succincte du fonctionnement d'un port parallèle. Pour plus de détails, voyez par exemple : "SYSTEMES A MICRO-PROCESSEURS" de D.J. DAVID chez EDITESTS.

Un port d'entrée-sortie c'est d'abord une adresse qui se comporte comme une adresse-mémoire : on peut y lire, par PEEK, ou y écrire, par POKE, une donnée de 8 bits ou octets. La différence avec une case mémoire ordinaire, c'est qu'il y a une correspondance entre chaque bit de cette pseudo-case mémoire et un fil qui sort de l'ORIC.

Connecteur"imprimante"vu depuis l'arrière de l'ORIC



"case"d'adresse 769 (301)

Pour le voir, faisons un essai. Branchez un voltmètre entre la broche 17 et une des broches paires (elles sont toutes à la masse). Vous devez trouver une tension voisine de 5 V.

C'est normal, si l'on regarde le contenu de la case d'adresse 769 - oui, c'est cette adresse-là qui intervient, par construction de l'ORIC - en faisant ?PEEK(769) on trouve 254, valeur dont le bit 7 est à 1.

Nous retenons donc bit à 1 = fil à + 5V et nous commençons à nous douter que bit à 0 = fil à 0V.

Pour le vérifier, on peut faire POKE 769,0 qui mettra bien tous les bits - dont le 7 - à 0 et remesurer la tension.

Las ! On trouve encore 5V. Nous serions-nous trompés ? Non ! Nous sommes victimes d'un tour que nous joue le système : il remet les bits à 1 après chaque instruction BASIC. Dans toutes les manips de ce genre, il faut s'attendre à des interactions parfois gênantes avec le système. Dans l' ORIC, à la différence des COMMODORE qui offrent un port parallèle entièrement libre pour l'utilisateur, le port sert aussi à l'imprimante et au décodage clavier; le système agit donc dessus de temps en temps.

Ici, cela ne nous gênera pas trop. Entrez et exécutez la ligne :

```
100 POKE 769,0 : GOTO 100
```

Là on voit l'aiguille du voltmètre osciller autour de 2,5V. Cela montre que notre POKE met bien la tension à 0 et que le système la met à 5. Le fait que les oscillations soient visibles montre la lenteur de BASIC.

Nous avons donc vu un premier mode d'utilisation du port dans lequel la donnée écrite à l'adresse du port impose une tension correspondante sur la broche extérieure (bit 0 = 0V ; bit 1 =5V). Ce mode s'appelle mode sortie.

Dans le second mode de fonc-

tionnement appelé entrée, c'est au contraire le monde extérieur qui impose une tension (0 ou 5V) sur la broche du port, et cette tension détermine le bit (0 ou 1) qui sera obtenu si on lit à l'adresse du port.

Mais comment décide-t-on du mode ? Il y a une deuxième case mémoire (dans notre cas d'adresse 771) dont chaque bit est en correspondance avec le bit de même numéro de la case 769. La case 769 s'appelle le registre donnée du port; la case 771 s'appelle le registre direction. Un bit 1 dans le registre direction entraîne que le bit correspondant fonctionne en mode sortie. Un bit 0 dans le registre direction entraîne que le bit correspondant fonctionne en entrée. Si nous faisons

```
? PEEK (771)
```

nous obtenons 255 ce qui montre que tous les bits du port fonctionnent en sortie, si l'on ne fait rien. C'est pour cela que notre expérience précédente a marché.

Faisons maintenant

```
POKE 771,0
```

Nous voyons alors quelque chose d'étrange : le clavier ne fonctionne plus; certaines touches sont inopérantes, d'autres donnent un caractère différent (Ex. RETURN donne *). C'est dû, là encore, au conflit entre le système et nous pour utiliser le port. Bon ! Eteignez puis allumez votre ORIC et entrez

```
10 POKE 771,0
20 ? PEEK (769) : GOTO 20
RUN
```

Il vient plein de 255. Ceci montre que laisser libre une broche en entrée est équivalent à lui imposer + 5V. Relions alors la broche 17 à la masse. Il s'imprime des 127. (CQFD).

L'ennui est que, pour reprendre le contrôle du clavier, il faut faire RESET, car le système a besoin d'avoir le port en sortie.

On va alors faire intervenir une nouvelle propriété du port parallèle qui va nous permettre de lire l'état du monde extérieur même si le port est en sortie. Entrez le programme d'une ligne :

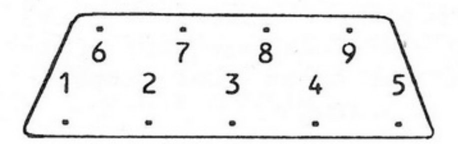
```
20 ? PEEK (769) : GOTO 20
```

Il s'imprime des 254 car le bit 0 est mis à 0 par le système. Relions la broche 17 à la masse : il vient des 126 donc l'état du bit 7 est bien lu à 0. Si nous voulons arrêter, nous pouvons faire CTRL C ce qui montre qu'on a gardé le contrôle du clavier.

Nous sommes donc maintenant capables de lire l'état de n'importe quel interrupteur branché entre une broche du port parallèle et la masse : si l'interrupteur est ouvert, on lit un bit 1; s'il est fermé, on lit un bit 0.

Le manche à balai

Un manche à balai est précisément un ensemble de 5 interrupteurs entre la masse et une broche. La plupart des constructeurs font des modèles compatibles, notamment ATARI et COMMODORE. Le brochage est montré ci-dessous :



Prise Cannon 9 broches

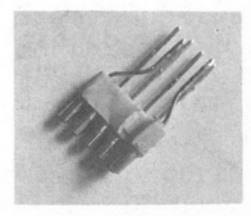
1 : NORD	2 : SUD
3 : OUEST	4 : EST
6 : FEU	8 : masse

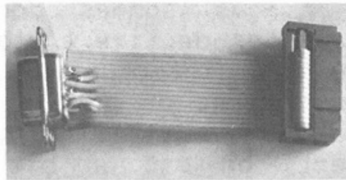
Pour connecter un manche à balai à votre ORIC, il vous suffit d'établir les liaisons :

Manche à balai	ORIC
1	15
2	13
3	11
4	9
6	17
8	10

Les numéros sont ceux des brochages des figures ci-dessus.

Nous avons réalisé avec des douilles à wrapper l'adaptateur représenté sur la photo ci-dessous. Bien sûr, il fait un peu "bricolage" et lorsqu'on le branche sur l'ORIC, il faut faire attention: sa broche à l'extrémité droite doit aller en 17 et non en 19.





Pour un montage plus professionnel, il faut prendre un connecteur pour câble plat 2x10 contacts 3M SCOTCHFLEX 3421 - 6000 et un connecteur Cannon D mâle, 9 contacts. L'un et l'autre coûte 20 à 25 F.

La SEDERMI vend le montage tout fait pour 65 FF TTC. Vous pouvez le commander au 28,rue Vicq d'Azir (référence: ORICJOY-1). Les logiciels LORICIELS qui font appel à un manche à balai sont compatibles avec ce montage.

Logiciel

Une fois la connexion faite, il faut un programme pour l'exploiter. Il résulte du tableau des connexions que les bits de la case 769 mis à zéro par les directions du manche sont :

Direction	Bit mis à 0	X pour test par (PEEK(769)ANDX)=0
NORD	6	64
SUD	5	32
OUEST	4	16
EST	3	8
FEU	7	128

Une manière astucieuse de procéder est de définir des fonctions logiques. Par exermple FNSUD(Z) vaudra -1 (vrai) s'il est vrai qu'on dirige le manche vers le sud et 0 sinon. On aura :

```
20 DEF FNSUD(Z)=(PEEK(769)AND32)=0
```

et les autres à l'avenant sauf que, pour le NORD, on ne peut appeler la fonction FNNORD qui contient le mot-clé OR !!! Merci BASIC !
On prend le nom FNNRD et le tour est joué !

Pour FEU, on a le choix entre:

```
50 DEF FNFEU(Z)=PEEK(769) 128
```

et

```
   DEF FNFEU(Z)=(PEEK(769)AND128)=0
```

Le programme qui suit est une esquisse de télécran basse résolution. Il dessine si on appuie sur FEU et laisse en blanc sinon .

```
10 DEF FNNRD(Z)=(PEEK(769)AND64)=0
20 DEF FNSUD(Z)=(PEEK(769)AND32)=0
30 DEF FNOUE(Z)=(PEEK(769)AND16)=0
40 DEF FNEST(Z)=(PEEK(769)AND 8)=0
50 DEF FNFEU(Z)= PEEK(769)   128
60 CLS:X= 20 :Y= 13 :PLOT X,Y,7
70 X = X + FNOUE(Z) - FNEST(Z)
80 Y = Y + FNNRD(Z) - FNSUD(Z)
90 IF FNFEU(Z) THEN PLOT X,Y,"*"
100 GOTO 20
```

Pour une version plus perfectionnée, voyez la cassette LA COMMODE.

Honoric de BALSA

Les secrets de la GP 80

Excellent rapport qualité-prix

1°)CARACTERES SEMI-GRAPHIQUES CBM:

Ils sont impossibles a obtenir directement (voir plus loin les possibilités en mode graphique). Il est donc opportun de placer quelques REM dans les programmes à lister, spécialement s'il y a des programmations du curseur.

2°)MINUSCULES:

On les obtient par les codes ASCII de 97 à 122. Avec des routines effectuant la translation de code on peut donc travailler en mode majuscule et obtenir les minuscules par SHIFT et vice-versa. Naturellement le texte à l'écran ou à l'imprimante n'a pas le même aspect!

3°)TABULATION:

De même que sur les imprimantes CBM l'instruction TAB a le même effet que SPC sur écran, c'est à dire un espacement à partir du dernier caractère imprimé (sur la ligne) et non à partir du bord de la feuille. Néanmoins on peut tabuler à partir du bord par la séquence suivante:

```
PRINT CHR$(16);CHR$(h);CHR$(b);
"CHAINE"
```

ou h et b sont les parties haute et basse respectivement du nombre d'espaces à partir du bord, avant le premier caractère de "CHAINE". Cela équivaudrait sur l'écran à:

```
PRINT TAB(10*h+b)"CHAINE"
```

avec la particularité que b est en modulo 16.

Remarque: les nombres h et b peuvent être transmis par le code écran des deux premiers caractères de la chaîne.

Exemple:

```
PRINTCHR$(16);"BDZZZ"
```

imprimera la chaîne ZZZ tabulée à 24 espaces du bord. 24 = 2*10+4 (2=code écran pour B, 4=code écran pour D).

4°)MODE GRAPHIQUE:

Le travail en mode graphique est plus simple et plus souple que sur les imprimantes CBM.

Sur la SEIKOSHA les caractères standards sont réalisés par une matrice 7x5. Il faut compter une colonne élémentaire pour séparer les caractères si bien que chaque ligne normale peut être décomposée en 80x6 = 480 colonnes élémentaires de 7 points chacune.

Pour programmer l'impression de chacune de ces colonnes élémentaires il faut d'abord entrer en mode graphique par une instruction PRINT CHR$(8).

Ensuite on procède comme suit: chaque point sur une colonne a un poids selon la position occupée:

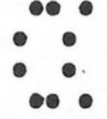
```
o - poids  1
o -   "    2
o -   "    4
o -   "    8
o -   "   16
o -   "   32
o -   "   64
```

On additionne les poids que l'on

veut imprimer et on y additionne 128. Exemples:

a) PRINTCHR$(8);CHR$(213) --> ●
o
213=128+1+4+16+64 ●
o
●
o
●

b)
```
DATA 8,176,200,200,176,200,128
FOR I = 1 TO 7
READ N
PRINT CHR$(N)
NEXT
```



Une bibliothèque de sous-programmes de ce type permet d'utiliser l'alphabet grec, les symboles mathématiques etc...

Remarque:

176 = code "O" SHIFT = ⌐
128 = ASC ("■")
200 = code "M" SHIFT = ◫

aussi l'instruction suivante a le même effet que la séquence précédente:

PRINT CHR$(8);"⌐◫◫⌐◫■"
(RVS SHIFT '@')

Malheureusement les codes 224 à 255 qui correspondent à RVS SHIFT, de SPC à "?" ne peuvent pas être transmis par ce procédé.

Il faut bien noter que l'on reste en mode graphique tant que on n'en sort pas par un PRINT CHR$(15) (caractères normaux) ou CHR$(14) (caractères double largeur).

Pendant le travail en ce mode l'interligne disparait automatiquement. D'autre part l'impression colonne par colonne sans avoir à faire des "paquets" correspondant à la matrice d'un caractère normal facilite grandement le tracé de courbes ou d'un dessin interprété point par point.

5°)TABULATION par colonnes élémentaires:

En mode graphique la tabulation décrite en 3°) fonctionne parfaitement mais elle est souvent trop grossière.

Il est possible de tabuler par colonnes élémentaires par l'instruction:

PRINTCHR$(27);CHR$(16);CHR$(h);CHR$(b)

ou h et b sont les parties haute et basse de l'adresse de la 1ère colonne élémentaire imprimée, cette adresse étant A = 256*h+b.

Exemple:

PRINTCHR$(27);CHR$(16);CHR$(1);CHR$(66);CHR$(193)
provoque l'impression de 2 points à la colonne élémentaire n:

1*256+66 = 322
(notez que 193 = 128+1+64).

Remarque: h et b peuvent être transmis par une chaîne de caractères mais h est lu modulo 64 (donc code écran) et b par son code ASCII.

Compte tenu de toutes les remarques

PRINTCHR$(27);CHR$(16);"AB♠"

produira le même effet d'imprimer deux points à la colonne 322. En effet:

code écran de A = 1 }
code ASCII de B = 66 } 256+66 = 322

code de "♠" = 193

Ce mode de tabulation fonctionne aussi en mode caractères.

6°)REPETITION:

Si l'on désire répéter la même impression sur un certain nombre de colonnes élémentaires,

PRINTCHR$(28);CHR$(n);CHR$(193)

provoquera l'impression des deux

points (code 193) sur n colonnes élémentaires consécutives. L'effet sera celui de deux lignes parallèles. Cette instruction ne marche qu'en mode graphique.

7°) DOUBLE LARGEUR:

On y entre par PRINT CHR$(14) et on en sort par CHR$(15) (normal) ou CHR$(8) (graphique). Elle double la largeur des caractères et des tabulations. Un retour chariot ne fait pas sortir de ce mode.

8°) FORMATAGE:

Toutes les instructions de formatage par un fichier ouvert à une adresse secondaire sont inopérantes. En fait l'adresse secondaire n'existe pas sur la GP 80.

9°) COMMANDES ET CODES ASCII CORRESPONDANTS

Description	Intitulé	code dec.	code hexa.
Entrée en mode graphique	BS	8	08
Nouvelle ligne	NL	10	0A
Retour charriot	RC	13	0D
Mode caractères double largeur	SO	14	0E
Mode caractères normaux	SI	15	0F
Tabulation	POS	16	10
Retour sans saut de ligne	DC4*	20	14
Ouverture tabulation élémentaire	ESC	27	1B
Repétition en mode graphique	FS	28	1C

* DEL sur Commodore.

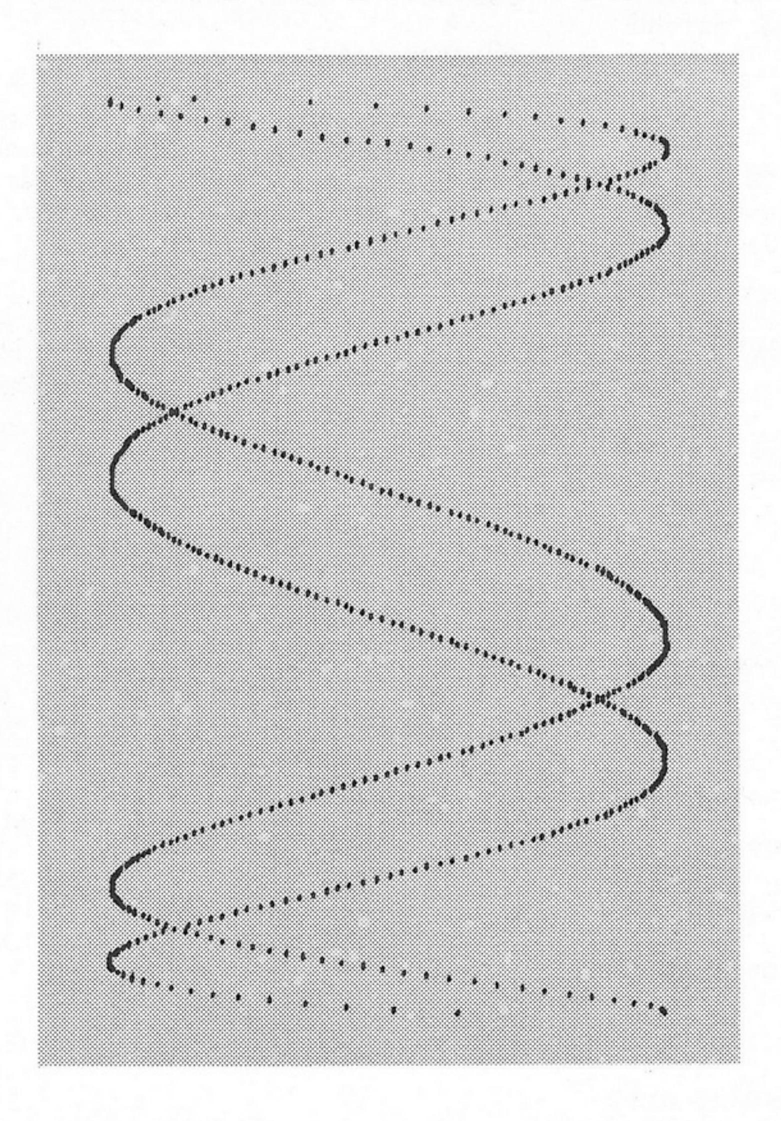
En conclusion voici un programme, qui peut être optimisé, avec le résultat obtenu pour des valeurs particulières, et qui illustre un certain nombre de possibilités de la SEIKOSHA, notamment en mode graphique.

P. CANALS

LISSAJOUS-IMPRIMANTE 11-6-82

```
 5 PRINT"LISSAJOUS:X=SINT;Y=SIN(KT+F)
 10 PRINT:PRINT:INPUT"K";K:INPUT"F";F
 15 OPEN1,4:CMD1,CHR$(14)
 20 PRINT"LISSAJOUS:X=SINNT;Y=SIN("K"*T+"F")":PRINT:PRINTCHR$(8):X=-1.0067
 25 B=.0068:FORL=1TO42:FORI=0TO6:X=X+B
 27 T=ATN(X/SQR(1-X*X))
 30 Y(I)=INT(100*SIN(K*T+F))+150:Z(I)=INT(100*SIN(K*(-T)+F))+150:NEXT
 50 PRINTCHR$(28);CHR$(49);CHR$(128);:FORI=50TO250:E=0:FORJ=0TO6
 55 IFY(J)=IORZ(J)=ITHENE=E+2^J
 60 NEXT:PRINTCHR$(E+128);:NEXTI:PRINTCHR$(128):NEXTL:PRINT#1:CLOSE1:GOTO5
READY.
```

LISSAJOUS:X=SINT,Y=SIN(5 *T+ .7)



*
* *

M. VAISSIERE de Montbeugny nous a, lui aussi, livré ses impressions sur la GP80 qui corroborent celles de M. CANALS.

Nous citons trois de ses exemples représentatifs:

```
10 OPEN1,4
20 PRINT#1,"BONJOUR ";CHR$(14);"COMMENT CA VA";CHR$(15);" BIEN MERCI"
30 PRINT#1,
40 CLOSE1
```

BONJOUR COMMENT CA VA BIEN MERCI

Le CHR$(15) n'est utile que pour annuler l'effet d'un autre CHR$, pour revenir en mode normal. En particulier, si on n'annule pas CHR$(14) par CHR$(15) et que l'on veut lister, le listing sera en caractères élargis.

```
10 OPEN1,4
20 PRINT#1,"BONJOUR ";CHR$(14);"COMMENT
CA VA";CHR$(15);" BIEN MERCI"
30 PRINT#1,
40 CLOSE1
```

Un beau panier de crabes:

```
10 A$=CHR$(128)+CHR$(128)+CHR$(128)+CHR$(128)+CHR$(128)
20 B$=CHR$(143)+CHR$(144)+CHR$(175)+CHR$(192)+CHR$(128)
30 C$=CHR$(128)+CHR$(136)+CHR$(248)+CHR$(136)+CHR$(128)
40 D$=CHR$(136)+CHR$(248)+CHR$(136)+CHR$(128)+CHR$(192)
50 E$=CHR$(175)+CHR$(144)+CHR$(143)+CHR$(128)+CHR$(128)
60 F$=CHR$(128)+CHR$(128)+CHR$(128)+CHR$(128)+CHR$(128)+CHR$(128)+CHR$(128)
61 G$=CHR$(144)+CHR$(144)+CHR$(200)+CHR$(164)+CHR$(162)
62 H$=CHR$(186)+CHR$(254)+CHR$(254)+CHR$(254)+CHR$(255)
63 I$=CHR$(191)+CHR$(157)+CHR$(137)+CHR$(227)+CHR$(227)
64 J$=CHR$(227)+CHR$(227)+CHR$(137)+CHR$(157)+CHR$(191)
65 K$=CHR$(254)+CHR$(254)+CHR$(254)+CHR$(250)+CHR$(186)
66 L$=CHR$(162)+CHR$(164)+CHR$(200)+CHR$(144)+CHR$(144)+CHR$(128)+CHR$(128)
71 M$=CHR$(136)+CHR$(225)+CHR$(144)+CHR$(136)+CHR$(132)
72 N$=CHR$(130)+CHR$(130)+CHR$(131)+CHR$(131)+CHR$(135)
73 O$=CHR$(135)+CHR$(135)+CHR$(135)+CHR$(135)+CHR$(135)
74 Q$=CHR$(131)+CHR$(131)+CHR$(131)+CHR$(130)+CHR$(132)
75 P$=CHR$(135)+CHR$(135)+CHR$(135)+CHR$(135)+CHR$(135)
76 R$=CHR$(136)+CHR$(144)+CHR$(225)+CHR$(129)+CHR$(130)+CHR$(128)+CHR$(128)
100 OPEN4,4:PRINT#4,CHR$(13),CHR$(15)
105 FORI=1TO14
110 PRINT#4,CHR$(8);A$+B$+C$+D$+E$+F$,
112 NEXT
113 PRINT#4,CHR$(13)
115 FORI=1TO14
120 PRINT#4,G$+H$+I$+J$+K$+L$,
122 NEXT:PRINT#4,CHR$(13)
123 FORI=1TO14
125 PRINT#4,M$+N$+O$+P$+Q$+R$,
126 NEXT
127 PRINT#4,
130 CLOSE4
```



J. VAISSIERE

Conclusion generale

Le fonctionnement de cette imprimante est particulièrement satisfaisant en regard de son prix. La seule limitation est l'impossibilité d'obtenir les caractères spéciaux Commodore dans les listings. Bien sûr, les imprimantes Commodore qui sont des SEIKOSHA adaptées (notamment la VIC 1515) n'ont pas cette limitation.

Tout ce banc d'essai est valable aussi pour la GP100 (qui est en 80 colonnes mais imprime plus large) et aussi - probablement - pour la GP40 (qui est en 40 colonnes). L'avantage de la GP100 est qu'elle n'exige pas le papier spécial étroit de la GP80.

P. CANALS
J. VAISSIERE

Bibliographie

"Le livre du VIC"
par Benoît MICHEL, éditions BCM.

A l'heure où les bouquins traitant du VIC commencent à se multiplier, on pourrait s'imaginer que celui-ci fasse double emploi. Eh bien, ce n'est pas le cas, car il constitue une mine de renseignements et de trucs pour celui qui désire approfondir ses connaissances et s'ouvrir des possibilités nouvelles. Il ne s'adresse donc pas aux débutants, mais celui qui aura fait le tour de son BASIC n'aura aucune peine à assimiler le contenu très dense de ses 250 pages. L'exposé est, en effet, accompagné de nombreux exemples courts, simples et bien choisis. Les points délicats font l'objet d'un programme bien documenté. Ainsi présentée, l'analyse du fonctionnement des VIA n'est pas du tout indigeste, comme c'est le cas dans "VIC REVEALED". L'auteur nous invite ensuite à découvrir le fonctionnement de l'interpréteur BASIC. Son expérience des ROM COMMODORE est appréciable lorsqu'il s'agit de décrire les routines les plus utiles. Enfin, chose peu courante, on trouve un chapitre sur les extensions matérielles où sont indiqués les composants nécessaires et même les schémas d'implantation. En annexe, on trouvera de nombreux programmes utilitaires et le listing d'un assembleur un peu plus puissant que "VICMON" . Je pense que ce programme rembourse l'achat du livre (120 F chez SIDEG).

Bref, j'ai été conquis. Un seul reproche: l'auteur-éditeur n'a pas amélioré sa reliure et une simple lecture transforme le livre en un amas de feuilles volantes.

H. LE MARCHAND

N.D.L.R. Bonne nouvelle : ce livre est maintenant facile à trouver en France, puisqu'il est diffusé par P.S.I.

ORIC 1 pour tous
par Jacques BOISGONTIER
et Sophie BREBION.

La découverte de l'ORIC
par Daniel-Jean DAVID

Voici deux ouvrages publiés à peu d'intervalle par les Editions du PSI, qui vont chacun à leur façon nous permettre enfin d'apprendre à nous servir correctement de notre cher ORIC. Oubliés, et le Manuel dont tous les bancs d'essai ont souligné les insuffisances, et le Guide dont il a été dit, ici même, ce qu'il fallait penser.

ORIC-1 pour tous pourrait être sous-titré L'ORIC par l'exemple. Ici en effet on ne trouvera qu'un strict minimum de texte, chaque instruction étant explicitée par un petit programme, souvent d'ailleurs accompagné de croquis et/ou d'un ordinogramme.

Des programmes un peu plus étoffés viennent, le cas échéant donner un exemple d'application plus large, notamment pour les instructions graphiques propres à l'ORIC. Et on notera avec satisfaction que dans ce domaine, les auteurs ont écrit nombre de petits programmes originaux fort plaisants.

La présentation d'ORIC-1 pour tous est très soignée, la typographie claire. Voilà un livre agréable à consulter.

Mais, pourquoi diable les auteurs ont-ils cru nécessaire de conserver les termes anglais "BACK-

GROUND" et "FOREGROUND" à propos des couleurs? Il est exact que leur traduction littérale "ARRIERE PLAN" et "AVANT PLAN" n'est pas satisfaisante dans le cas présent. Les termes "FOND" et "ENCRE" sont tout-à-fait acceptables et admis. Est-ce trop demander d'y penser pour une nouvelle édition ? Laquelle pourra avantageusement être débarassée de nombreuses petites coquilles peu graves, mais agaçantes.

Dans "La découverte de l'ORIC" la démarche est bien différente. D.J. DAVID a eu l'ambition non seulement d'apprendre à utiliser l'ORIC, mais aussi et surtout de faire comprendre le BASIC d'abord, les finesses de l'ORIC ensuite.

Très clair, rédigé dans un excellent français - si peu courant dans la littérature micro-informatique - la Découverte de l'ORIC s'adresse à un double public.

Au débutant d'abord, qui a tout à apprendre. A l'amateur confirmé également qui veut connaître l'ORIC qu'il vient d'acquérir et qui pourra lui, aborder directement la deuxième partie du livre.

Les explications sont illustrées par de multiples programmes, abondamment commentés et que le lecteur est vivement incité à expérimenter.

Mais la sanction des connaissances acquises ne sera véritablement donnée que par la résolution - exacte - des exercices proposés. Un effort qui en vaut la peine.

Une fois ce livre refermé - et l'ORIC découvert - il reste encore bien des points à approfondir.
Nul doute qu'ils feront l'objet d'un La Pratique de l'ORIC, certainement déjà en chantier, n'est-ce-pas D.J. DAVID ?

Honoric de BALSA

Graphiques de gestion
par D. CHARPENTIER, chez Masson.

Ce livre est un recueil de sous-programmes de tracé fonctionnant sur système CBM 8000 auquel on connecte un traceur HP 7470 A par le bus IEEE. Il est un excellent prolongement du chapitre 5 de "La pratique du CBM" Vol.1, de D.J. DAVID chez PSI ou on connecte un traceur HP 7225. Les commandes des deux traceurs sont analogues, mais le 7470, plus moderne, est moins cher et a 4 couleurs.

Le principal mérite de cet ouvrage est qu'il couvre des applications de gestion et non des applications scientifiques pour lesquelles il existe tant d'ouvrages (de toutes façons, les sous-programmes donnés sont utilisables en scientifique). C'est dire qu'il abonde en diagrammes de statistiques des ventes, d'évolution du chiffre d'affaires, etc.,

Il exploite les couleurs dans des diagrammes de répartition en "fromage", ou dans des courbes en spirale. Tout ceci est fort bien fait.

Pierre-Etienne THALBERG

The COMMODORE 64 Programmer's Reference Guide

The COMMODORE 64 User Manual

Pas de commune mesure entre ces deux livres. L'User Manual n'est rien d'autre que - traduit- le manuel livré avec la machine.
(N.B. Si! il existe en français et vous DEVEZ l'obtenir auprès de votre distributeur - Parlez éventuellement de la loi qui stipule que tout matériel vendu en France doit avoir une notice en français). Donc, c'est tout juste le manuel pour commencer.

Avec le Reference Guide, c'est tout autre chose. Il contient notamment une description du système avec les adresses du noyau (le Kernal) et comment les utiliser, ainsi qu'un guide sur le langage machine et l'utilisation des périphériques.

Mais le chapitre qui était le plus nécessaire est celui sur les graphiques (le 3). Il explique tous

les modes graphiques notamment la haute résolution et les sprites. En revanche, le chapitre sur les sons nous a déçu : il ne donne guère plus d'éléments que l'user manual. La plus grande critique à adresser à ce livre indispensable est la difficulté à se le procurer. Aux dernières nouvelles, PROCEP ne l'avait pas encore en quantités.

Pierre-Etienne THALBERG

Visa pour ORIC
par F. BLANC
et F. NORMANT.
éd. SORACOM (diffusion ASN)

Ce petit livre de 62 pages, moitié du format de LA COMMODE est vendu 40 F, ce qui est assez cher pour la surface.

Il contient néanmoins un certain nombre d'adresses utiles sur le système de l'ORIC, sujet sur lequel les informations sont très rares. Un certain nombre d'explications données sont utiles, d'autres sont très naïves. Il y a aussi un certain nombre d'erreurs.

Les adresses des routines qui effectuent les fonctions BASIC sont données sans explication : les auteurs indiquent qu"'il serait trop long de décrire tous les paramètres ... il vous suffit de désassembler la ROM à partir de l'adresse donnée pour les trouver". Là, les auteurs envoient le bouchon un peu loin: si c'était si facile, ils l'auraient fait.

Daniel-Jean DAVID

Réservé aux Abonnés

CONNAISSEZ VOTRE NUMERO D'ABONNEMENT :

L'étiquette d'envoi de votre numéro porte, en tête, un numéro très important. C'est, en fait, par lui que nous vous connaissons. Gardez ce numéro, il vous est indispensable dans vos relations avec nous. Il est utile de le rappeler dans toutes circonstances et c'est indispensable pour :
- réabonnement,
- changement d'adresse,
- services réservés aux abonnés (Ex: photocopies),
- réclamations

REABONNEMENT :
N'oubliez pas de rappeler votre numéro. Cela vous évitera de recevoir une lettre de relance alors que vous venez de vous réabonner.

CHANGEMENT D'ADRESSE EN COURS D'ABONNEMENT :
Il est impératif de citer votre numéro, sinon la prise en compte de votre changement risque d'être retardée, donc un exemplaire risque de vous être envoyé à votre ancienne adresse. D'autre part, il faut maintenant joindre 10 F en timbres à tout changement d'adresse.

RECLAMATIONS :
Toute réclamation pour numéro non parvenu doit rappeler votre numéro d'abonné. Elle doit être formulée par lettre non recommandée et pas par téléphone ni par visite au 28 rue Vicq d'Azir, où il n'y a AUCUN numéro.

PHOTOCOPIES :
La Commode va mettre en place différents services réservés aux abonnés (Nous pensons même à un service de consultation technique).
Dès maintenant, nous fournissons photocopies d'articles de numéros épuisés.
Tarif : 10 F par article + 2 F par page. Commandez par lettre à
La Commode 28, rue Vicq-d'Azir 75010 PARIS

Communiqués de Presse

Nous avons le plaisir de vous annoncer l'ouverture de LOGIC-STORE, un nouveau point de vente en micro-informatique.

Un de plus...direz-vous !

Attention ! ...LOGIC-STORE est le premier magasin consacré à :

La micro-informatique familiale

Notre mission :

- Sensibiliser le public aux applications de l'ordinateur à la maison,
- Proposer une gamme de produits (THOMSON TO 7, ATARI 400 et 800, GENIE III et IV) spécifiquement grand public,
- Présenter autour de ces matériels un choix de revues, livres et logiciels de jeux éducatifs et de gestion,
- Assurer aux acheteurs de nos produits une assistance pour leurs premiers pas dans le monde de la micro-informatique familiale.

LOGIC-STORE, 39, rue de Lancry
75010 PARIS
(Contact : Bernard TAHON,
tél. : 206.72.28)

*
* *

Depuis le mois de décembre, RADIO-MAYENNE (96,6 MHz), première radio départementale de RADIO-FRANCE, existant depuis le 16 juin 1980, ouvre ses antennes aux adeptes de la micro-informatique.

Ainsi, chaque jeudi soir, à 23 heures, Yves DERISBOURG vous propose de diffuser vos programmes de micro-ordinateurs enregistrés sur cassettes.

Une bourse d'échange de programmes informatiques s'ouvre pour tous les amateurs de la région (LE MANS, ANGERS, RENNES, ALENCON et, bien sûr, toute la MAYENNE) qui peuvent adresser leurs programmes originaux enregistrés sur cassettes. (Tous systèmes : ZX 81, COMMODORE, APPLE etc..,etc..,) à l'adresse suivante :

Yves DERISBOURG
RADIO-MAYENNE
9,rue du Lieutenant
53000 LAVAL

Pour tous renseignements, appeler dans la semaine Yves DERISBOURG au

56 - 38 - 88 par le 43

A VOS CLAVIERS.....
A VOS CASSETTES .....

*
* *

J.C.R fort de plusieurs expériences réussies d'implantations de magasins franchisés, a pensé offrir aux magasins de Hi-Fi/Vidéo un rayon Micro-Informatique complet "clé en main".

Ces points de vente que nous avons baptisés "Micro-Boutique J.C.R" offrent non seulement le matériel, mais aussi les meubles, la PLV, en fait toute l'infrastructure commerciale à la vente de produits Micro-Informatique.

Grâce à son volume d'achat important, J.C.R bénéficie des meilleurs prix du marché et offre aux exploitants des "Micro-Boutiques J.C.R." des conditions de travail

optimum, pour une bonne rentabilité de la surface de vente. Le choix des produits, les relations avec les fournisseurs, le stockage du matériel, l'élaboration des campagnes publicitaires... autant de problèmes pris en charge et résolus par J.C.R.

J.C.R. Electronique
56 - 58, rue Notre-Dame-de-Lorette
75009 PARIS

*
* *

La Société TEREL
4, rue Mademoiselle
78000 VERSAILLES
commercialise des matériels et logiciels optimisés pour la gestion des cabinets de petite et moyenne importance.

*
* *

SILICONE VALLEE DEMARRE DANS LA MICRO

Créé il y a quatre ans pour distribuer des composants électroniques, des kits et de la mesure dans l'Ouest de la France, SILICONE VALLEE vient de décider de diffuser des micro-ordinateurs domestiques dans ses trois points de vente, ORIC, COMMODORE VIC, 64 etc.,

<u>Points de vente SILICONE VALLEE</u>

NANTES : 87, quai de la Fosse
Tél (40) 73.21.67
5, rue Lekain
Tél (40) 89.71.26

ANGERS : 7, rue Boisnet
Tél (41) 88.13.98

* *
*

EPS ouvre une nouvelle division, EPS Systèmes, qui se charge de la vente des microordinateurs VIC 20 et C64.

EPS SYSTEMES, 45 rue des Petites Ecuries,75010 PARIS

Tél (1) 523.35.30

EPS SYSTEMES, 25 rue Ambroise Croizat, 78280 GUYANCOURT

Tél (3) 043.51.90

* *
*

J.C.R. s'agrandit !

Vous connaissez sûrement notre magasin de Micro-Informatique de la rue Notre-Dame de Lorette qui est un des plus réputés de Paris.

Vous avez appris en Décembre 1982 la création de notre première franchise "J.C.R. Boutique" à Marseille.

Aujourd'hui nous vous informons avec plaisir de l'ouverture de notre seconde "J.C.R. Boutique" à Lyon :

313, rue Garibaldi (7ème).

Comme dans les autres "J.C.R Boutique", vous y trouverez toutes les grandes marques de l'informatique : COMMODORE, APPLE, SHARP, THOMSON, ORIC, SEIKOSHA, VICTOR LAMBDA, EPSON, etc., mais aussi le plus vaste choix de logiciels, périphériques et accessoires.

J.C.R.ELECTRONIQUE
56-58,rue Notre-Dame-de-Lorette
75009 PARIS

Tél. (1) 282.19.80

Petites Annonces et Clubs

Vds VIC - 20 + DATA 7 + 16 Ko + PS 2000 + 6 CARTOUCHES (SUPER LANDER, AVENGER, SARGON II, OMEGA RACE, etc..,) + 2 livres +Nombreux jeux K7 + PROGRAMMER'S AID + BONZO + AUTOFORMATION . VENDU: 5.000 F.

Renaud PATALOWSKI
72/74, av. Parmentier 75011 PARIS
Tél. 357.95.91

Je possède un VIC 20 et j'échange sur cassette : un programme SPACE MATH et BIORHYTHM COMPATIBILITY contre : programme gestion de fiches et petit commentaire.

B. MARGUERITTE
20, promenade Newton,
30000 NIMES



Dans le n° 8 de LA COMMODE est paru mon article "A LA RECHERCHE DES ROUTINES PERDUES". J'ai relevé, page 54, 4 erreurs d'impression :

1 - CHAPITRE UTILISATION, 2ème paragraphe :

Au lieu de : ici A0 B3 A2 18 ...
Lire : A0 83 A2 18 ...

M. Pierre FREMION, 63, rue de l'Amiral Mouchez, 75013 PARIS aimerait correspondre avec d'autres utilisateurs de VIC 20 pour échanger des informations, particulièrement sur le langage machine.

=-=-=-=-=-=-=-=

Club

THEORIC est un club pas tout-à-fait comme les autres, dont le but est double :

Initier et favoriser la pratique de micro-ordinateurs, et créer un lien réel de rencontre entre les générations, par le biais du micro-ordinateur.

THEORIC
4, Petite Place
B.P 198
78001 VERSAILLES CEDEX
Tel. H.B (3) 951.29.30
Soir (3) 021.63.24
(3) 916.14.46

2 - CHAPITRE UTILISATION, 6ème paragrahe :

Au lieu de : Le programme s'arrêterait immanquablement à $02B7
Lire : Le programme s'arrêterait immanquablement à $027B (et alors seulement, il devient aisé de "deviner pourquoi" !)

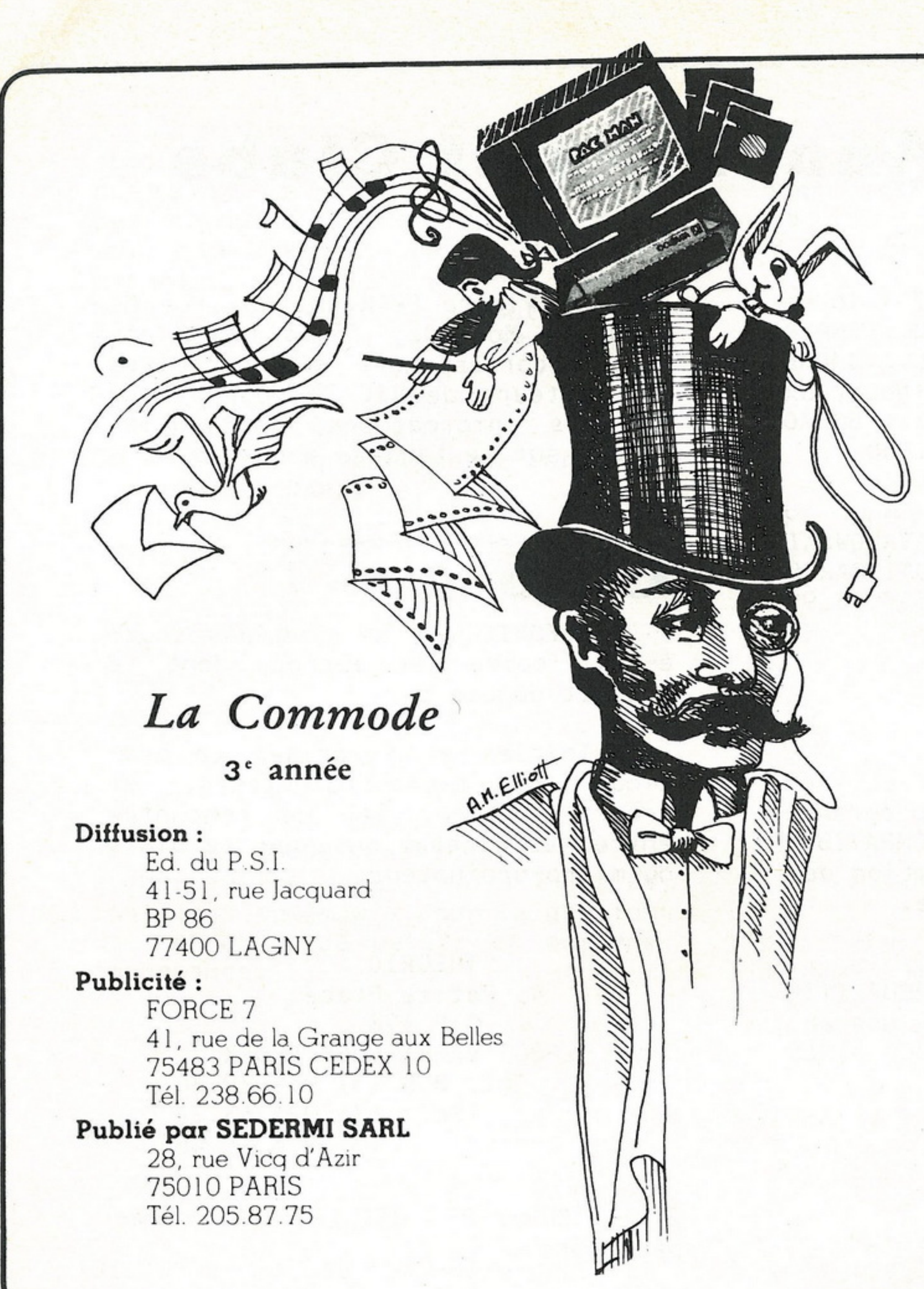
3 - CHAPITRE UTILISATION, Rem. 4 :

Dans la ligne 110, il manque un "= D2" après AND PEEK (I + 2)

4 - CHAPITRE RESULTATS SUR CBM 4000 P, 2ème paragraphe :

Au lieu de :
("SCROLLING UP"): SYS 58217 ($E379)
Lire: " " ($E369)

J. PIERRAT
78000 VERSAILLES



La Commode

3ᵉ année

Diffusion :
Ed. du P.S.I.
41-51, rue Jacquard
BP 86
77400 LAGNY

Publicité :
FORCE 7
41, rue de la Grange aux Belles
75483 PARIS CEDEX 10
Tél. 238.66.10

Publié par SEDERMI SARL
28, rue Vicq d'Azir
75010 PARIS
Tél. 205.87.75

Rédaction

Rédacteur en chef :
Daniel TRECOURT

Rédacteur en chef adjoint :
Marc VAN CRAEYNEST

Chefs de rubriques :
Jacques COUTHURES
Jean DELAVILLE
Jean-Luc DESCHAMPS
Pierre-Etienne THALBERG
François VULQUIN

OOO

Directeur de la publication :
Daniel-Jean DAVID

Rédaction-vente-abonnements :
28, rue Vicq d'Azir
75010 PARIS
Tél. 205.87.75

Bulletin d'abonnement

A envoyer à La Commode
28, rue Vicq d'Azir - 75010 PARIS

Je désire m'abonner à la Commode (1 an, 4 numéros) à partir du numéro ....

NOM : .... PRÉNOM : ....

Adresse : ....

....

Signature

Règlement 150 F à l'ordre de SEDERMI :

☐ CCP ☐ CB ☐ ESPECES

(étranger : 160 FF supplément avion : 40 FF)

Réabonnement ☐
Indiquez si possible votre ancien numéro d'abonné

Imprimerie RL Promotion, 95880 Enghien. Imprimé en France, Directeur de la publication Daniel-Jean David
Dépôt légal : décembre 1983 - Numéro de commission paritaire : en cours ISSN 0292-5079



Super Concours

Trouvez la clef du mystère de KIKEKANKOI

1 magnétoscope

cassettes à gagner !

Nombreux autres titres
Pour recevoir la liste complète des cassettes Loriciels, retournez-nous vite une enveloppe affranchie à votre adresse.

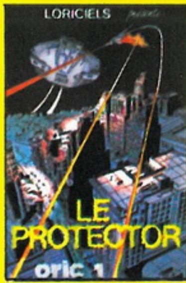
Bulletin de participation, et extrait du règlement insérés dans la cassette, à nous retourner avant le 15 février 1984 minuit, tampon de la poste faisant foi. Règlement déposé chez Maître Pacalon, huissier de justice à Paris.

ORIC 1 48 K



A - Le mystère de KIKEKANKOI : 180 F : un très grand jeu d'aventure pour l'ORIC 1. Vous devrez faire preuve d'un immense courage pour franchir les obstacles et libérer la belle prisonnière. 55 tableaux différents.

ORIC 1 48 K



B - LE PROTECTOR: 95 F
Un superbe jeu composé de trois tableaux, écrit en langage machine. Au premier tableau, vous survolez une ville dont vous avez la responsabilité. Hélas, vous n'êtes pas seul.

ORIC 1 48 K



C - ORION : 95 F
Un jeu dangereux pour la santé ! Vous ne pourrez plus vous arrêter d'y jouer. Magnifique jeu d'arcades en trois actes. Mouches robotisées, base spatiale, vaisseau d'attaque...
Langage machine

ORIC 1 48 K



D - LE MANOIR DU Dr GENIUS : 140 F
Exceptionnel ! Armez vous de patience, ce jeu d'aventures, totalement graphique, vous fera passer des heures inoubliables dans le Manoir plein de mystères et de surprises.

ORIC 1 48 K



E - CROCKY : 120F
Enfin une superbe version en langage machine du célèbre jeu des gloutons poursuivant les fantômes.

TOUS NOS LORICIELS SONT GARANTIS 5 ANS

Vous avez écrit des logiciels pour micro ordinateurs si vous voulez être édités contactez-nous.

ORIC 1 48 K



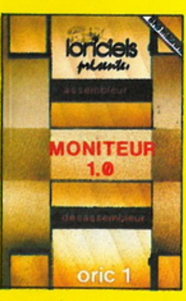
F - HU* BERT : 120F
En langage machine, un jeu superbe sonore et graphique. En sautant de cube en cube, le HU* BERT change leur couleur mais doit aussi éviter de nombreuses créatures étranges.
Un jeu où il vous faudra un grand sens de l'équilibre.

ORIC 1 48 K



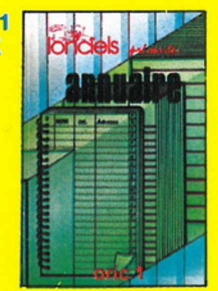
G - GASTRONON : 95 F
Un jeu complètement fou de huit tableaux en langage machine.
Il faut détruire avec une base spatiale, des choucroutes, des cornichons, des chopes de bière

ORIC 1 48 K



H- MONITEUR : 140 F
De nombreuses fonctions
100 % langage machine

ORIC 1 48 K



I - ANNUAIRE : 140 F
Gestion de fichier multi-critères

A VENIR, nombreux logiciels pour ZX - ORIC, COMMODORE Etc...



17, rue Lamandé 75017 PARIS
Tél. : (1) 627.43.59

Vente par correspondance exclusivement

Bon de commande
à envoyer à Loriciels
17 rue Lamandé 75017 Paris - Tél. 627.43.59

Nom .... Prénom ....
Adresse ....
Ville .... C.P. ....
Tél. .... Date ....

Signature :

Livraison sous 48 h à réception de commande dans la limite des stocks disponibles.

Référence	A	B	C	D	E	F	G	H	I
Prix unit. TTC	180	95	95	140	120	120	95	140	140
Quantité									

....

Frais de port* ....

PRIX TOTAL TTC : ....

*Frais de port 10 F par cassette pour tout envoi hors France Métropolitaine

Paiement à adresser avec le bon de commande
ci-joint : chèque bancaire ☐
CCP ☐

CI



SPID VOUS OFFRE SA 1re SELECTION DE LOGICIELS.
Une sélection mondiale de 392 programmes pour : APPLE - ATARI - IBM - CBM - TRS - Sharp PC EPSON - Sinclair ZX81 et Spectrum ORIC 1 - Victor Lambda - Dragon.
Automne 1983
GUIDE DES LOGICIELS
UTILITAIRES
SCIENTIFIQUES
FAMILIAUX
JEUX
EDUCATIFS
SPID micro
APPLE • ATARI • COMMODORE VIC • DRAGON • EPSON • ORIC • IBM • SHARP PC • SINCLAIR ZX et SPECTRUM • TRS • VICTOR LAMBDA • etc.
Vous y trouverez :
• Une description de chaque programme accompagnée (autant que possible) d'une photo d'écran et de l'emballage d'origine.
• Le prix moyennement constaté de chaque programme.
• La liste des distributeurs auprès de qui vous pourrez vous approvisionner.
SPID approvisionne votre distributeur et garantit pendant 1 an votre logiciel contre tout défaut de fabrication.
GRATUIT CHEZ LES DISTRIBUTEURS SPID
Exigez le Label de Qualité
SPID micro
GÉNÉRATION!
Demandez-le chez votre distributeur micro-informatique habituel ou à SPID par correspondance en renvoyant le bon ci-dessous.
SPID. 39, rue Victor Massé 75009 Paris - Tél. 281.20.02
Je désire recevoir le guide des Logiciels SPID. Je joins 5 F en timbre en participation aux frais d'expédition.
Nom
Prénom
Adresse
Code Postal
Ville